ÆTHER WIRE & LOCATION, INC.

> ## Low-Power, Miniature, Distributed Position Location and Communication Devices Using Ultra-Wideband, Nonsinusoidal Communication Technology

Semi-Annual Technical Report

Contract J-FBI-94-058

Prepared for:

Advanced Research Projects Agency /
Federal Bureau of Investigation

Wireless, Adaptive, Mobile Information Systems
Principal Investigators Meeting

July 1995

Principal Investigators: Robert Fleming
                                Cherie Kushner

Table of Contents

PROJECT GOALS

Æther Wire's long term goal is the development of *coin-sized* devices that are capable of localization to *centimeter* accuracy over *kilometer* distances. These "Localizers" will be able to operate within a network of millions of other units in a local area, and users will be able to enter and leave the network seamlessly and transparently. Ultimately, these localizers will be able to operate for up to a year on a watch-sized battery, or longer if augmented by solar power. The overall goal of this ARPA sponsored project is the development of pager-sized units powered by AAA-sized cells that are capable of localization to *sub-meter* accuracy over kilometer distances in networks of up to a few hundred Localizers.

BACKGROUND

Our research effort started with the goal of developing small, low power transceivers that can be used for position location and low data-rate communication. Position location can be determined by sharing range information within a network of transceivers. Pairs of transceivers resolve their separation by cooperatively exchanging an electromagnetic signal. The accuracy of this range determination is a function of the bandwidth of the exchanged signal. With conventional sinewave technology, the bandwidth of the signal relative to the carrier frequency is very small — at most a few percent using spread spectrum. However, it is possible to transmit and receive electromagnetic **impulses** which have a relative bandwidth approaching 100%. This "nonsinusoidal" radiation is currently being used for anti-stealth and ground-probing radar, under the more common heading of ultra-wideband or impulse radar.

Nonsinusoidal radiation has unique advantages when used, at many orders of magnitude less power than radar, for communication and cooperative ranging:

- Transceivers can be made very small (*i.e.* coin size), low power, and low cost because the electronics can be completely integrated in CMOS without any inductive components.
- The antennas can be equally small, and can be driven directly by CMOS, because they are non-resonant and current-mode.
- Ultra-wideband / nonsinusoidal signals form a shadow spectrum which can coexist and does not interfere with the sinewave spectrum. The transmitted power is spread over such a large bandwidth that the amount of power in any narrow frequency band is very small.
- The good features of spread spectrum are shared, including multipath immunity, tolerance of interference from other radio sources, and inherent privacy from eavesdropping (low probability of intercept).
- Ultra-wideband / nonsinusoidal signals have very good penetrating capabilities, probably due to their relatively large bandwidth. Localizers can operate within buildings, urban areas, or forests.
- Centimeter-level accuracy in determining range is possible without using expensive microwave (GaAs MMIC) technology because nonsinusoidal signals have very large relative bandwidth, and the circuits can be clocked at much lower frequencies.
- The cooperative nature of this technology allows for accurate ranging *without* the need for extremely accurate and stable (and expensive) master clocks to synchronize the system.

The combination of communication and position location capability within devices that are totally integrated, and essentially "throw-away," opens up a host of applications, especially for monitoring large numbers of sensors or objects dispersed over an area. Properly tasked and

1

distributed, they can serve as extensions of the senses of both people and machines into their environment, and can transfer information both for perception and for control.

Portable Command Unit.
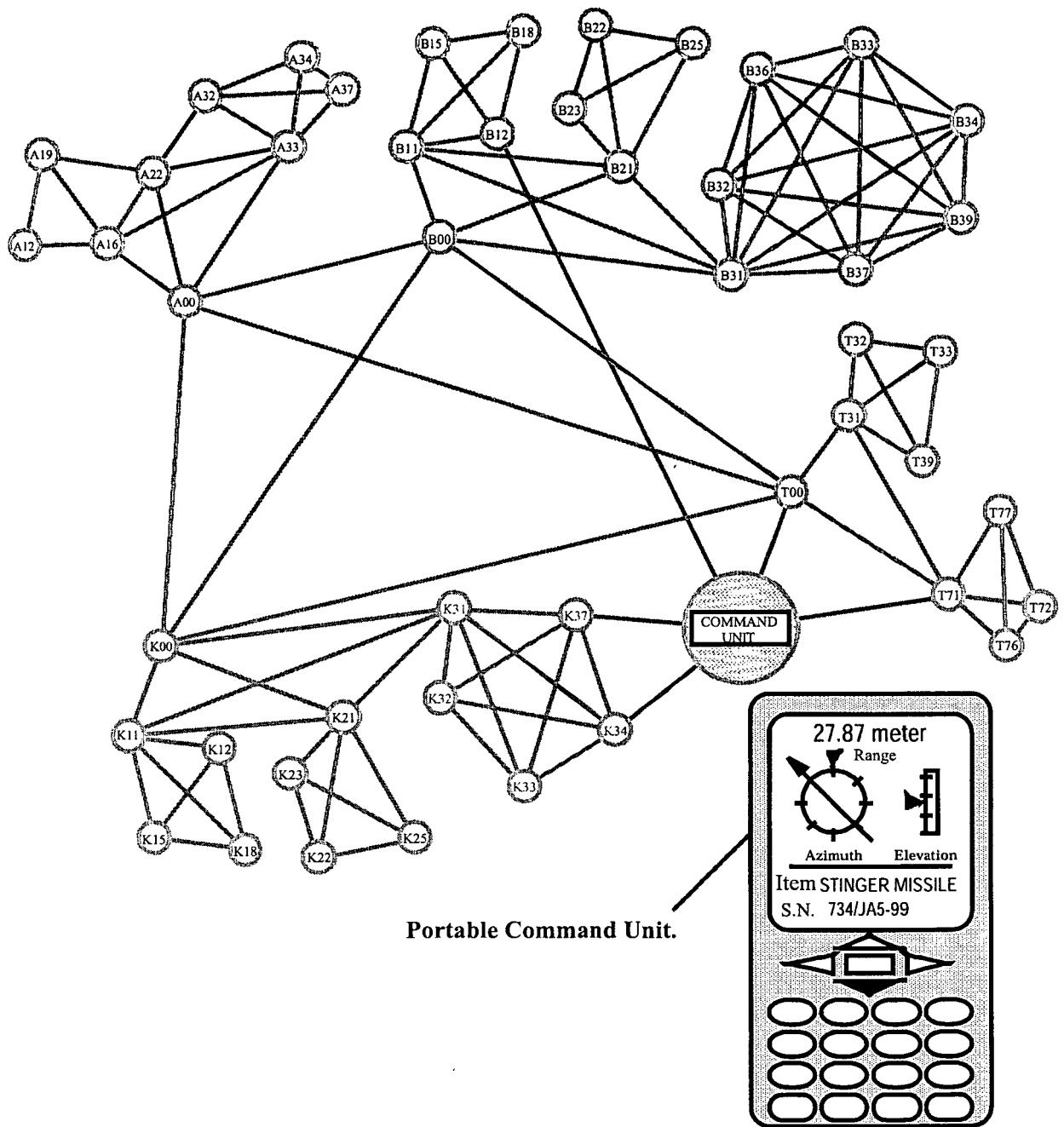
FIGURE 1. This diagram illustrates an AWL Localizer-based volumetric inventory system using a large network of Localizers attached to inventory items, where each Localizer determines the range to every other Localizer and then shares the information with members of the network. To allow many localizers to operate in a given area, the system architecture is set up with many branches and nodes. The portable command unit can access the network from anywhere and send queries for any item on the network from anywhere within or around the network. Once the user enters the identifier for the inventory item desired, the command unit queries the network and gives the user azimuth, elevation and distance to the item from its current position.

2

MILITARY APPLICATIONS

The ability of commanders to see through the fog of war has always made the difference between victory and defeat. Military strategists talk of the OODA cycle; observation, orientation, decision and action. At all levels of command, the faster a soldier or commander can get through the OODA cycle, the higher the probability of his success. The future battlefield will be dominated by the force that has superior information technology.

The Gulf War demonstrated the power and value of satellite-based remote sensing technology as a tool to help commanders see through the fog of war, but it also showed its limitations. The coalition forces still required dangerous low-altitude air reconnaissance and ground reconnaissance missions. In addition, satellite remote sensing technology was still centralized and had to be distributed from a centralized command authority — like the mainframe computers found in large corporations during the 1960's. In the future battlefield, information technology needs to be distributed with all soldiers networked with their units, their weapons systems and each other, so that information flows laterally through the network and up the chain of command as well as down. This "mesh" of information will allow the entire army to get through the OODA cycle faster.

The Gulf War also highlighted the need for effective IFF (Identify Friend or Foe) technology, and the ability to locate and identify friendly units. Three quarters of the coalition force vehicles that were damaged or destroyed were hit by "friendly fire".

Logistics management was another weakness highlighted in the Gulf War. Combat trains and logistics have historically been a vulnerability of fighting forces — a vulnerability that has multiplied as the daily logistics consumption of a combat division has increased from 100 tons per day in W.W.I to 300 tons per day in W.W.II to over 1000 tons per day today. The Geopolitical climate that the U.S. operates in almost guarantees that the future battlefield will be characterized by very long combat trains, and the specialized nature of modern weapon systems will preclude local sources of munitions and supplies more and more. This means that modern armies will increasingly need effective means of tracking and identifying critical supplies.

The increase in information technology on the battlefield[1] will also hasten the trend to more decentralization of command and control structures and lessened reliance on large combat platforms. Much as networking in the corporate environment has lessened the power of centralized, mainframe information systems and increased the ability of unit managers to make timely business decisions, the increased networking of the battlefield will allow small unit leaders to make faster, more effective decisions with local information rather than waiting for information and direction from higher command. This mesh architecture for C3I (Command Control Communications and Information) makes the future army more survivable and less vulnerable to "decapitation".

The small size, low power consumption, range, accuracy, encryption and anti-jam features of the Localizer make it an ideal platform for a variety of future battlefield applications:

- **IFF**: Localizers are an ideal means of providing individual soldiers with IFF capability. In addition, the ability to group sets of Localizers into "families" associated with a squad, fighting vehicle or team makes it harder for enemy units to infiltrate using captured Localizers. Furthermore, Localizers can be integrated with other sensors for monitoring the soldier's state (e.g. blood pressure, heart rate, body temperature, etc.), and can signal distress

if any of these factors go outside of safe limits. Localizers would also enhance C3I for the unit commanders and the theater commander.

- **Logistics Control**: Localizers act as bar code identification tags that not only can be read from a kilometer away, but can give the precise location of the logistics item. As Localizers get small and inexpensive enough, they can be placed on individual weapons and even individual munitions items. These could also be linked with the C3I system to give commanders real-time information on logistics usage and proximity to battlefield units.
- **Ground Surveillance**: Localizers combined with microsensors (e.g. gas sensors, vibration sensors, etc.) can be the heart of a field-scatterable sensor array. Such an array could identify the size and composition of an enemy ground force (e.g. number of vehicles, weight of vehicles, gas or diesel, etc.). The precise knowledge of time and location within a network of Localizers allows a synthetic phased array antenna to be formed; one of its uses could be to uplink sensor information to satellites. The transmission could be nonsinusoidal, or small satellite transceivers could be combined with Localizers for sinusoidal transmission.
- **Mine Clearing**: Localizers combined with microsensors and small AGV's (autonomous ground vehicles) could be used as a mine clearing force. Small, inexpensive, robotic vehicles, such as those proposed by Rodney Brooks[2,3], combined with Localizers to ensure full geographic coverage, could clear an area of mines and unexpended munitions. The low cost of the Localizers and the robots would make any loss of units in the mesh inconsequential in terms of systems operating cost.
- **Smart Minefields**: Field scatterable minefields have the advantage of rapidly mining a large area. They have the disadvantage of being difficult to clear once hostilities have ended. Combining mines with Localizers would allow a friendly force to easily locate the small, non-metallic scatterable mines, and effectively clear the minefield.

Technology by itself does not win battles, but an innovative combination of technology with applications and tactics can give an overwhelming advantage to a well-organized and well-trained force. The combination of telegraphy, railroads and repeating rifles changed the rules of warfare in the late 19th century, and the combination of aircraft and internal combustion engines changed the rules of warfare in the early 20th century. We believe that Localizers can form the core of an information technology revolution that will change warfighting tactics much as the internal combustion engine did earlier this century.

COMMERCIAL APPLICATIONS

Like the future battlefield, information is increasingly a mission critical resource for businesses. Businesses are also going away from a centralized MIS (Management Information System) to a more decentralized client / server system. PDA's (Personal Digital Assistants) are a platform to make the workforce more decentralized and more mobile. Position location technology is becoming necessary for inventory control as just-in-time manufacturing puts an increasing strain on existing inventory control techniques. Position location technology is increasingly becoming a recognized need for providing rapid and timely information to the mobile workforce.

In consumer applications, consumers are increasingly relying on embedded information technology in the products they buy — everything from watches to kitchen appliances to automobiles.

The small size, low power consumption, range, and potential of low-cost mass-manufacture of the Localizer make it an ideal platform for a variety of consumer and commercial applications:

- **Personal Location**: Personal Localizers can be used for a variety of applications, from safety devices used by Police and Fire personnel, to recreational safety units (hunters, hikers and skiers), to consumer units (child locators, car locators, purse locators, etc.).
- **Inventory Control**: Localizers act as bar code identification tags that not only can be read from a kilometer away, but can give the precise location of the inventory item. As Localizers get small and inexpensive enough, they can be placed on even relatively low cost retail items. These could also be linked with the MIS system to give real-time information on purchase patterns and stocking levels.
- **Smart Highways**: Localizers can be used as the heart of an IVHS (intelligent vehicle highway system)[4]. Localizers placed along side of a road can be used as guideposts for vehicles being autonomously guided along a highway. These Localizers can also communicate with Localizers in other vehicles to provide local intelligence rather than trying to manage an IVHS centrally (much as individual birds in a flock avoid collisions on a local basis without centralized direction).
- **Machine Control**: Localizers combined with earthmoving vehicles could be used to automate construction operations. Localizers combined with small, inexpensive, robotic vehicles, such as those proposed by Rodney Brooks, could be used for infrastructure inspection (nuclear plants, bridges, sewers, etc.). Localizers would also make possible small, robotic house cleaning robots that could live under a couch and reliably clean the full carpet area.
- **Smart Homes**: Combining Localizers with home appliances such as TVs, ovens, lamps, etc. could make possible smart homes that turn on and off the correct appliances by knowing both the location of the various residents and the location of various home items with respect to those residents.

MAJOR ACCOMPLISHMENTS

HARDWARE

We have designed, fabricated and are testing a transmitter antenna driver chip (Fig. 2). We have just submitted for fabrication a chip (Fig. 3) with the remaining subsystems for completion of transceivers that can demonstrate sub-meter range accuracy over 100 meter distances in real time. The major components in this chip include:

- A 200 MHz phase-locked loop, in low noise logic;
- All three stages of the Real Time Clock, with the high speed stages in low noise logic, including event registers and comparators for both transmit and receive;
- A code sequence generator each for transmit and receive, which includes a 25-stage Linear Feedback Shift Register, a run-length counter, and a doublet generator, all in low noise logic;
- 32 Time-Integrating Correlator phases; each phase has a reference code latch, code/signal multiplier, integrator, offset correction, sample-and-hold, and output mux;
- An 800 MHz broadband differential receiver antenna amplifier with 60 dB gain.

The production of these chips was made possible by the invention of a new analog circuit development and testing methodology, one that has allowed us to fabricate one or more test chips per month since February.

SOFTWARE

- Upgraded "lab bench" Localizer demonstration to do software impulse train synchronization.
- Developed a Simulator that allows us to exercise and debug the actual Localizer software for synchronization and ranging.
- Programmed the control and measurement system for testing our chips.
- Wrote program for interfacing between different CAD tools.
- Evaluated several automatic placement/routing packages.

MILESTONES

The major milestone for the next six months is the integration of the chips and software we have developed with a commercial processor, and to construct a pager-sized technology demonstrator unit for delivery to ARPA. The tasks we need to perform for this include:

- Adapting the current test system for the transmitter and receiver chips in fabrication;
- Testing and revising the transmitter and receiver chips;
- Porting of synchronization, ranging, and networking protocols from our simulator;
- Coding and debugging neural networks to extend the detection range;
- Design and debugging of the processor board for the pager-sized units.

With each new revision of our transmit and receiver chips, we will put more circuits on chip. These will include:

- Circuits that we have deferred developing because their functions could be done using commercial parts, including an A/D converter, crystal oscillator, and voltage references;
- Revised versions of our receiver amplifier to improve the sensitivity, noise performance, and dynamic range;
- Floating gate circuits for adaptively canceling input offsets and for tuning frequency response (versus the dynamic adjustment via DAC's which is now implemented).
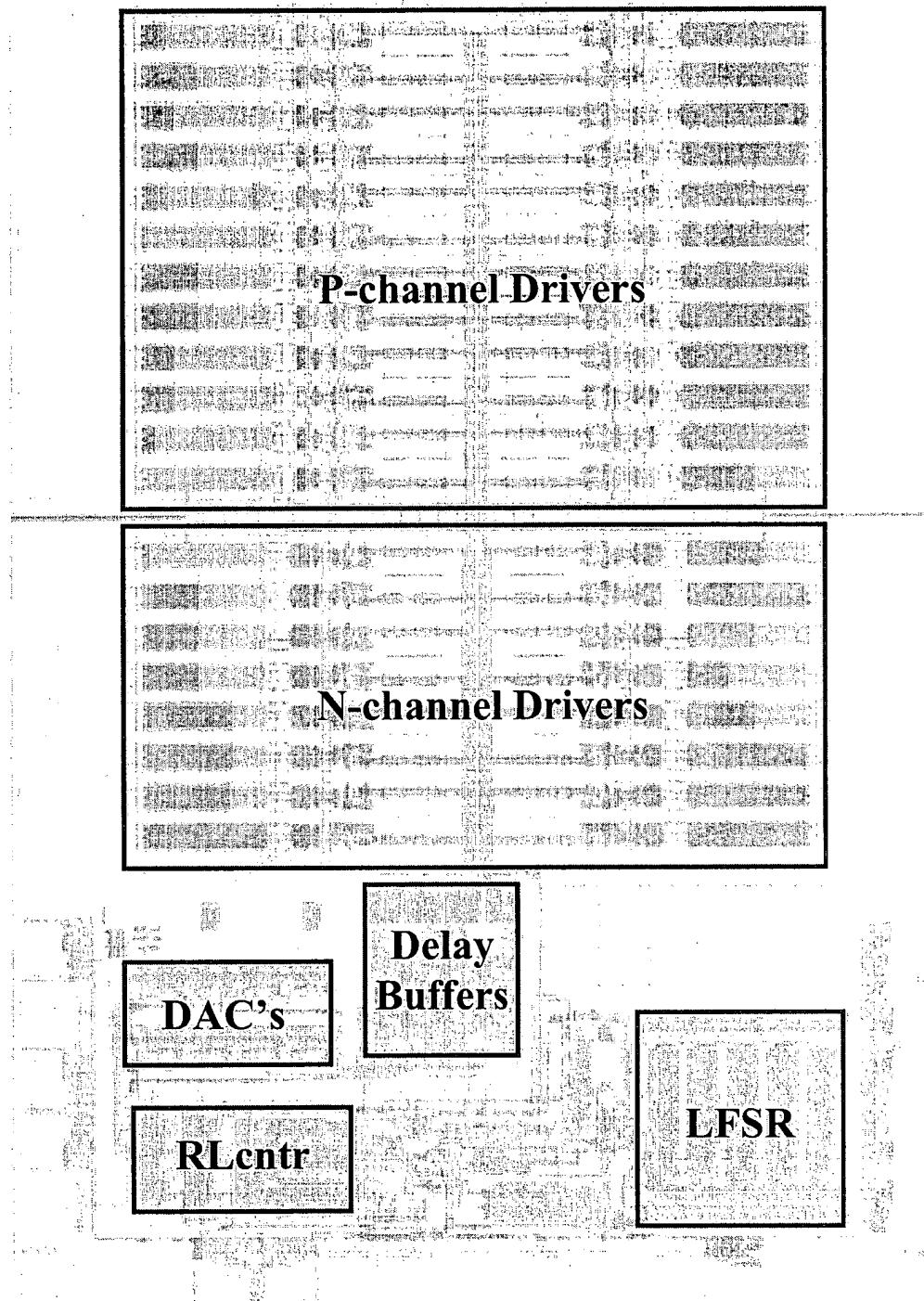
FIGURE 2. Transmitter Driver chip (**nstest11**) can switch 2 amps in 500 ps across the transmit antenna. The N-channel and P-channel transistors on both sides form an H-bridge configuration. There is an adjustable delay for both the positive and negative edges of the signals going to all 4 legs of the H-bridge. The output power is controlled by enabling any combination of 8-12 output sections. Multiple bond pads minimize the inductance connecting to the transmit antenna. This chip is driven by the Receiver/coordinator chip (**aether3**), but it also includes a code sequence generator for stand-alone testing.
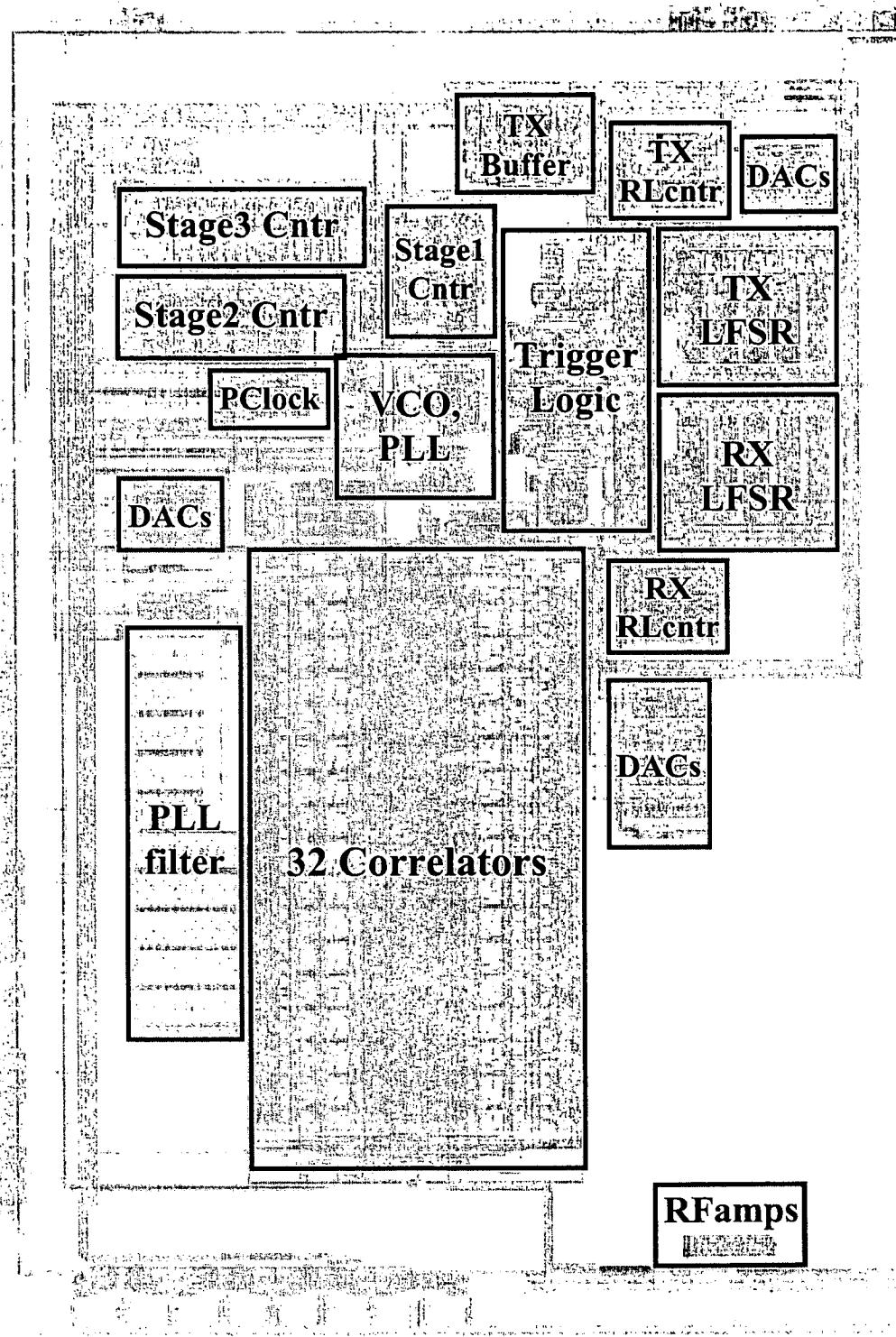
FIGURE 3. Receiver/coordinator chip (**aether3**) includes all the subsystems for reception of ultra-wideband signals, and drives the Transmitter Driver chip (**nstest11**) for transmission. The major components are a 200 MHz phase-locked loop, a multi-stage Real Time Clock for triggering transmit and receive events, dual code sequence generators for both transmit and receive, an 800 MHz broadband receiver antenna amplifier with 60dB gain, and 32 Time Integrating Correlator phases/integrators for detecting long code sequences.

DETAILED PROGRESS REPORT

SYSTEM OVERVIEW

The following system overview is reprinted for reference and for those new to this technology.

Our communication and position location technology is based on the transmission of *coded sequences of Gaussian impulses*, and selective reception using *correlation*. The transmitted signal is generated by applying current steps through a Large-Current Radiator antenna. An impulse is launched when the current is turned **on** or is turned **off**. The code sequences are pseudo-random codes like those used for direct-sequence spread spectrum systems such as GPS. Different sequences provide separate channels roughly equivalent to frequency bands. Correlation is used to discriminate a particular code sequence from other signals, both nonsinusoidal and sinewave frequency-based. For communication, the simplest modulation of code sequences is "antipodal" modulation: either a given code sequence or its inverse is sent to represent one bit of information.

The basic principle Localizers use for position location is *cooperative ranging*. Localizers cooperate in pairs to determine their separation, and five or more Localizers share information to determine the 3-dimensional relative location of each. The distance between two Localizers is determined by measuring the round-trip transit time for a signal and multiplying by the speed of light.

Other localization systems give absolute position on the geoid (*i.e.* GPS), location relative to fixed beacons (*e.g.* LORAN), or location relative to a starting point (*i.e.* inertial platforms). For a host of applications, what is really desired is location within a building or an area, or location relative to other people or objects, whether moving or stationary. Most localization systems allow autonomous position location, yet this information must be communicated by a separate mechanism in order to be shared. Knowing one's latitude and longitude is useless without additional information such as a map.

Ultra-Wideband / Nonsinusoidal Signals

Using the transit time of a signal to gauge distance requires an accurate measurement of the arrival time of the signal. The more sharply defined a signal is in time, the more spread out the signal is in the frequency domain. This is true for nonsinusoidal impulses as well as for the pulse-modulated sinewaves used in conventional radar. In either case, to measure distance to centimeter-level accuracy requires approximately 1 GHz of bandwidth (or more). The difference is that most of the energy of a nonsinusoidal 1 ns wide Gaussian impulse is spread over frequencies below 1 GHz (Fig. 4), whereas pulse-modulated sinewaves require a carrier frequency of at least 30–60 GHz to get this bandwidth. Expensive microwave (GaAs MMIC) technology is needed for sinusoidal transmission at a carrier frequency where it is possible to have sufficient bandwidth. Nonsinusoidal impulses are baseband signals, which means there is *no carrier frequency*.

The development of the ultra-wideband Large-Current Radiator (LCR) antenna by Dr. Henning Harmuth[5] has made it possible to radiate nanosecond wide impulses with inexpensive CMOS chips. The LCR is a current-mode antenna which radiates outwards from

the surface of a flat square conductor. The only restriction on its size is that it cannot be made larger than the equivalent width of the impulse being transmitted.

Applying a step change in the current through an LCR causes an impulse to be radiated, since radiation power launched is proportional to the square of the *derivative* of current flow. The impulse is narrower and has more radiated power when the current can be changed more quickly. The polarity of the impulse is determined by the sign of the derivative of current. Thus, turning **off** the current through an LCR generates an impulse which has the opposite polarity of the impulse generated when the current is turned **on**.
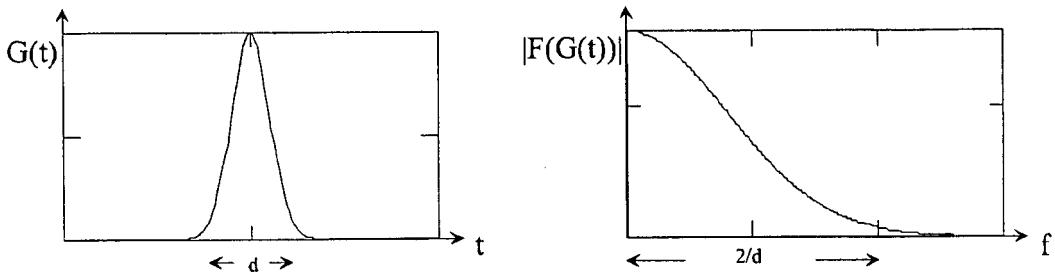

FIGURE 4. A Gaussian impulse of width **d** and its amplitude spectrum. Note that if **d** = 1 ns, most of the energy in the frequency domain is below 1/**d** = 1 GHz.

CODING

A *coded sequence* of Gaussian impulses means a series of electromagnetic impulses, each with the electric field vector pointing in one of two opposite directions. Correlators are used by Localizers to extract the signal from noise, even when the signal amplitude is less than the noise amplitude. The output of a correlator is a function of the relative shift between two signals. When the shift of the input signal matches that of the reference code, a correlation peak is produced. A signal spread out in time can thus be time-compressed to the resolution of a single impulse.

Ideally, codes should be chosen so that the correlation of a code sequence against itself will have a single peak, making it easy to determine when the proper sequence has arrived. Maximal Sequence codes and Complementary codes approach this ideal. The *cross-correlation* of one code sequence with a different sequence should *not have correlation peaks* in order for multiple Localizers to operate at the same time. Fortunately, families of codes exist with tens of thousands of members that have both good autocorrelation and good cross-correlation properties.

DOUBLETS

A series of impulses can be launched by stepping the current through the transmit antenna up or down. Unfortunately, consecutive impulses of the same polarity require stepping the antenna current higher and higher. Launching an arbitrary sequence of impulses can be very power consumptive, because the running difference in the number of positive and negative impulses determines the "DC" current, and this can become very large with almost all codes. The minimum total current occurs when, starting from zero current, a negative impulse immediately follows a positive impulse, and *vice versa*. We refer to such pairs of impulses as "doublets" (Fig. 5). Doublets can be generated from a single supply by using switching circuits to control the direction of current flow through the transmit antenna.
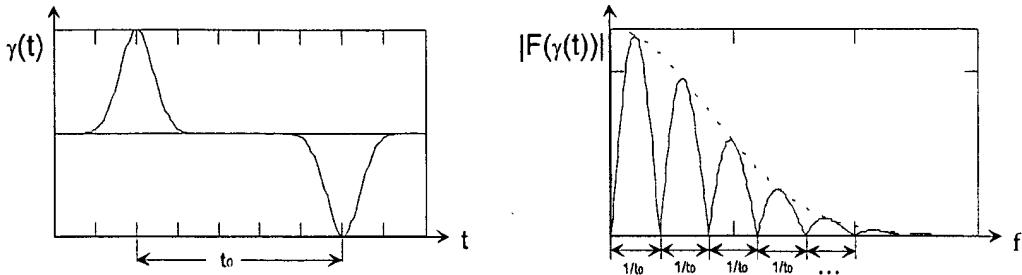
FIGURE 5. A "doublet" with impulse separation t_0 and its amplitude spectrum. For $t_0 = 5$ ns, the nulls are spaced 200 MHz apart with a null at 0 Hz (*i.e.* DC).

We have discovered that doublets can be used in a manner that both obviates the apparent restriction on having arbitrary code sequences, and that can be easily detected. For each bit in any code sequence, we generate a doublet which starts with either a positive impulse or a negative impulse. An example is shown in Figure 6. (In spread spectrum terminology, a doublet is our "**chip**".) If the autocorrelation of a sequence of impulses has a single correlation peak, then the autocorrelation of the same sequence encoded using doublets has a central peak bracketed by two negative peaks (Fig. 10). As discussed later, this complex pattern is much easier to recognize than a single peak, especially when the signal is contaminated with considerable noise.

An additional advantage of using doublet-encoded sequences is that the choices of the time separation between impulses in a doublet and the time separation between doublets allows the frequency spectrum to be manipulated (Fig. 5). For instance, nulls can often be placed at frequencies which harbor high intensity narrowband interference.
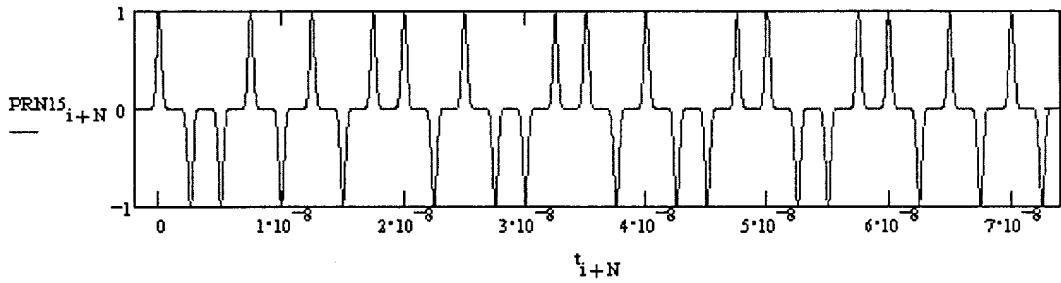

FIGURE 6. The 15-bit Maximal Sequence "100011011010111" encoded using doublets. Here the impulses are 1 ns wide and separated by 5 ns. Thus, most of the energy is below 1 GHz, with nulls every 200 MHz.

TIME-INTEGRATING CORRELATOR

The usual method for implementing correlators "slides" the analog input signal past the reference code sequence. The input signal is delayed, and taps weighted (+1, Ø, -1) by the reference code sequence are taken at successive delay stages. This Sliding Correlator (or matched filter) topology requires some sort of memory for the input signal. This may take the form of digital memory for the output of an analog-to-digital converter or a tapped delay line. Delay lines may be discrete, such as charge coupled devices (CCD's), or analog. Analog delay may take the form of coax, optical fiber, glass rods, surface acoustic wave (SAW) devices, all-pass filters, etc. All of these are unsuited for our purposes. ADC's and CCD's are too slow, and CCD's are also too power consumptive, SAW's have limited

11

programmability, and active delays such as all-pass filters cannot handle long codes. The rest cannot be inexpensively integrated.

We have chosen to use the *dual form* of the usual Sliding Correlator implementation, known as a Time-Integrating Correlator[6] (TIC). The reference code sequence is shifted past the changing analog input signal and the product of the code and signal is summed in a set of analog integrators (Fig. 7). The output of each integrator represents a different alignment ("phase") between the reference code sequence and the input signal (Fig. 8). The outputs of the TIC correspond nearly to the sampled output of a Sliding Correlator as a function of time.

The chief advantage of a TIC is that all of the difficulties of achieving precise and distortionless analog delay are replaced by the simple task of delaying a digital code sequence. The only limitations on the length of the code sequence are the stability of the timebase and the quality of the integrators. In fact, code sequences a million long have been used for satellite ranging. The disadvantage of a TIC is that a separate integrator is needed for what represents one sample of the output of a Sliding Correlator.
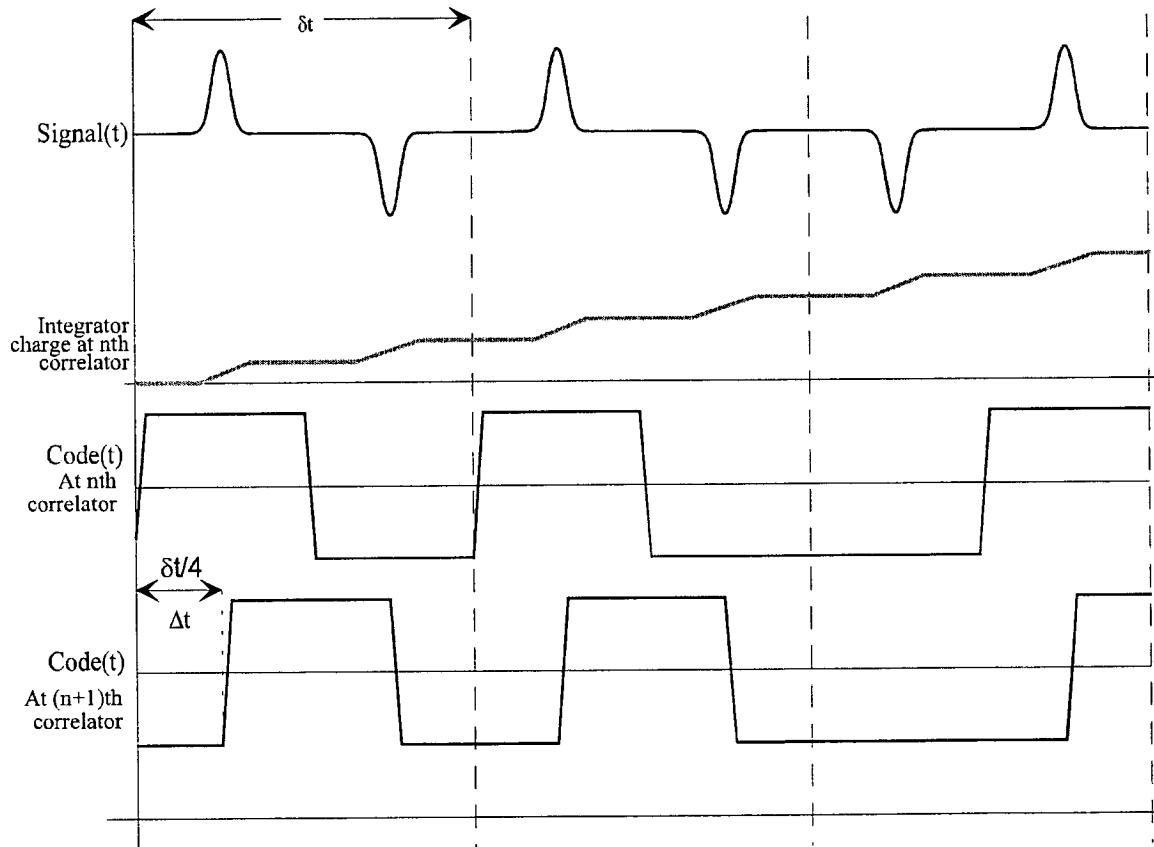
FIGURE 7. This diagram illustrates the operation of a Time-Integrating Correlator. The top row shows the received code sequence "110" encoded using impulse doublets. The second row shows the accumulation of charge in the nth integrator. The third row represents the +1/-1 values of the reference code sequence. The fourth row shows the overlap of the integration periods of the nth integrator with the (n+1)th integrator.

12
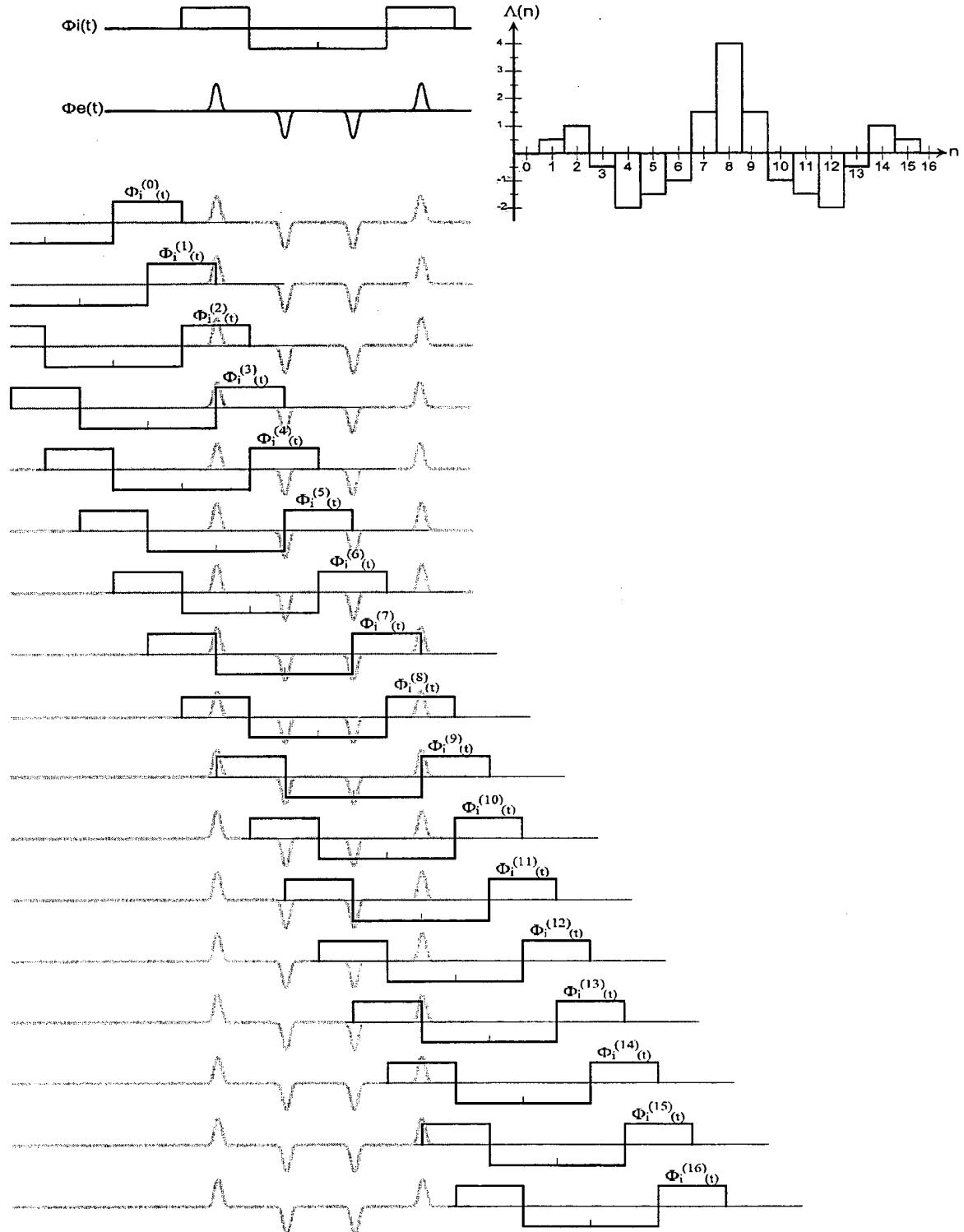
FIGURE 8. This diagram illustrates how the outputs of 17 integrator phases are generated by different alignments of the reference code $\Phi_i(t)$ with the received signal $\Phi_e(t)$. In this example, the integration period equals the impulse period, and adjacent integrator phases are overlapped by half of the impulse period.

With multiple integrators, a Time-Integrating Correlator (TIC) captures only a fraction of what would be the output of a Sliding Correlator. A continuously-operating Sliding Correlator can detect the arrival of a code sequence on the fly. A TIC needs to know roughly when to look, so as to position its window of integrator phases. Acquiring this knowledge is known as *synchronization*. The process can take time, but once synchronization is achieved, a TIC consumes much less power than a Sliding Correlator. Synchronization can be maintained by scheduling communication exchanges frequently enough to compensate for any drift in the relative clock rates among a group of Localizers.

DETECTION

A key feature of a Time-Integrating Correlator is that the process of detecting a correlation peak does *not* have to be done in real time. The correlation results are saved in the outputs of the integrator phases, which we then digitize. The complex TIC *patterns* that result can subsequently be analyzed with a microprocessor over a substantial period of time. This allows sophisticated recognition processes to be used such as neural nets and maximum entropy.

TIME DOMAIN FILTERING

For receiving a nonsinusoidal signal, a Time-Integrating Correlator *integrates* the product of the code and received signal over a period which is less than or equal to the separation of the transmitted impulses. The width of the integration period can be chosen for different purposes. When the integration period is as wide as the separation between impulses, the received impulses cannot fall outside the integration period for any integrator phase. If the integrator phases are also non-overlapping, then the set of integrator phases spans the largest time window, which is best for synchronization. When the integration period is reduced, the signal-to-noise ratio is improved, because less noise is integrated where there is no signal (Fig. 9).

Thus, this time domain filtering can be used, when the position of the received impulses is known quite accurately, to "mask out" noise where no signal is present. This works to increase signal-to-noise ratio until the integration window is approximately the size of the impulses and is straddling them.

14
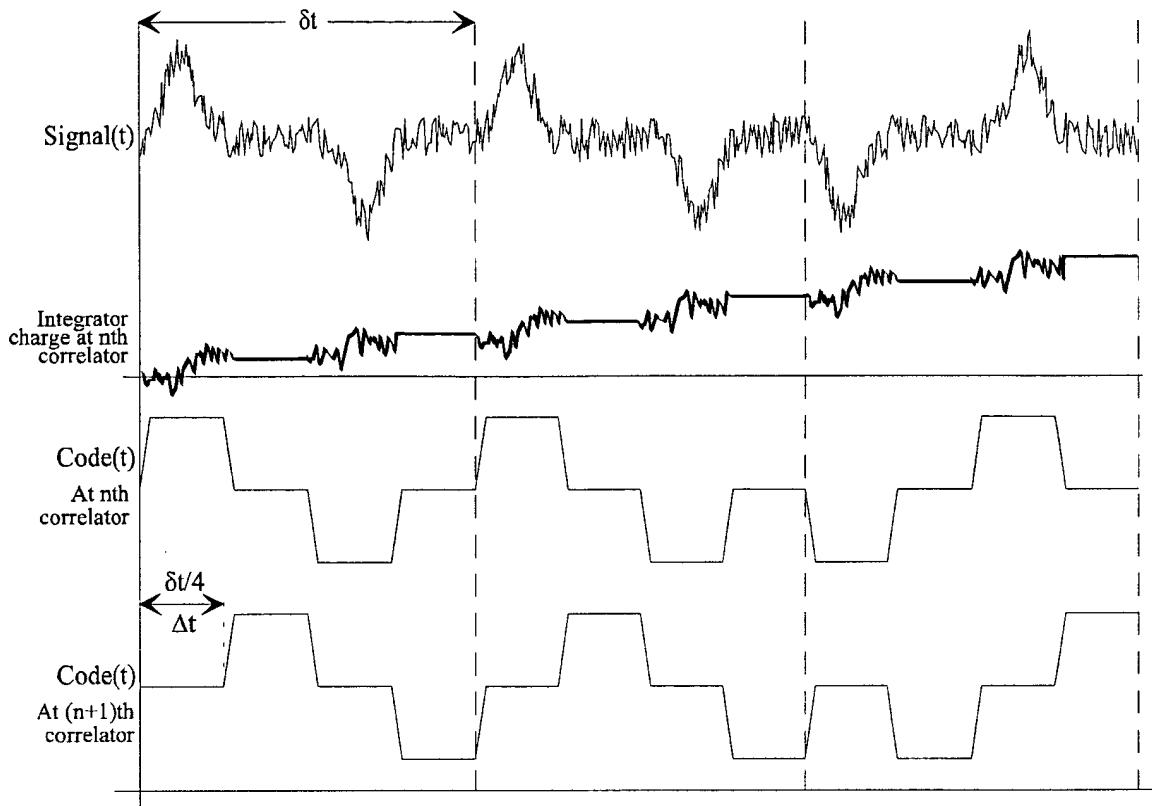
FIGURE 9. This diagram illustrates the operation of a Time-Integrating Correlator (TIC) where the integration period is half of the impulse period. The top row shows the received signal with noise. The second row shows the accumulation of charge in the nth integrator. The third row represents the +1/0/-1 values of the reference code sequence. The fourth row shows the reference code sequence for the (n+1)th integrator. Note how the input signal is effectively masked out when the reference is zero, with no loss in the signal contribution.

COMPLEX TIC PATTERNS FOR HIGH-SENSITIVITY TIME MEASUREMENTS

The pattern formed by the output phases of the Time-Integrating Correlator can be made very sensitive to slight shifts in the alignment of the received signal versus the reference code. This is achieved when the integration period for the TIC is as wide as the impulse period and the adjacent phases overlap by one half of the impulse period, as shown in Figure 7. This figure also illustrates the fact that the TIC performs the cross-correlation of two related but different signals — the received signal, consisting of Gaussian impulses, and the reference code which has a rectangular pulse for each impulse in the received signal. The continuous time cross-correlation of these two signals is shown as the trace labeled $\Lambda(t)$ in the left-hand column of Figure 10. The outputs of the TIC are shown as the trace labeled $\Lambda(n)$ in the right-hand column. The values of $\Lambda(n)$ correspond to *samples* of $\Lambda(t)$ separated by the spacing of adjacent phases of the TIC. The sample points are shown as delta functions in the left-hand column.

The output phases of the TIC which are the most sensitive to time shifts are those that correspond to samples of $\Lambda(t)$ where it is changing most rapidly. In Figure 10, these are the two outputs to either side of the central correlation peak in the $\Lambda(n)$ trace as well as the other "odd" numbered samples, where the central peak is "even". Also, the fact that the central

15

peak is essentially constant over a broad range of time shifts is important, since it serves as a reference level for automatic gain control functions.
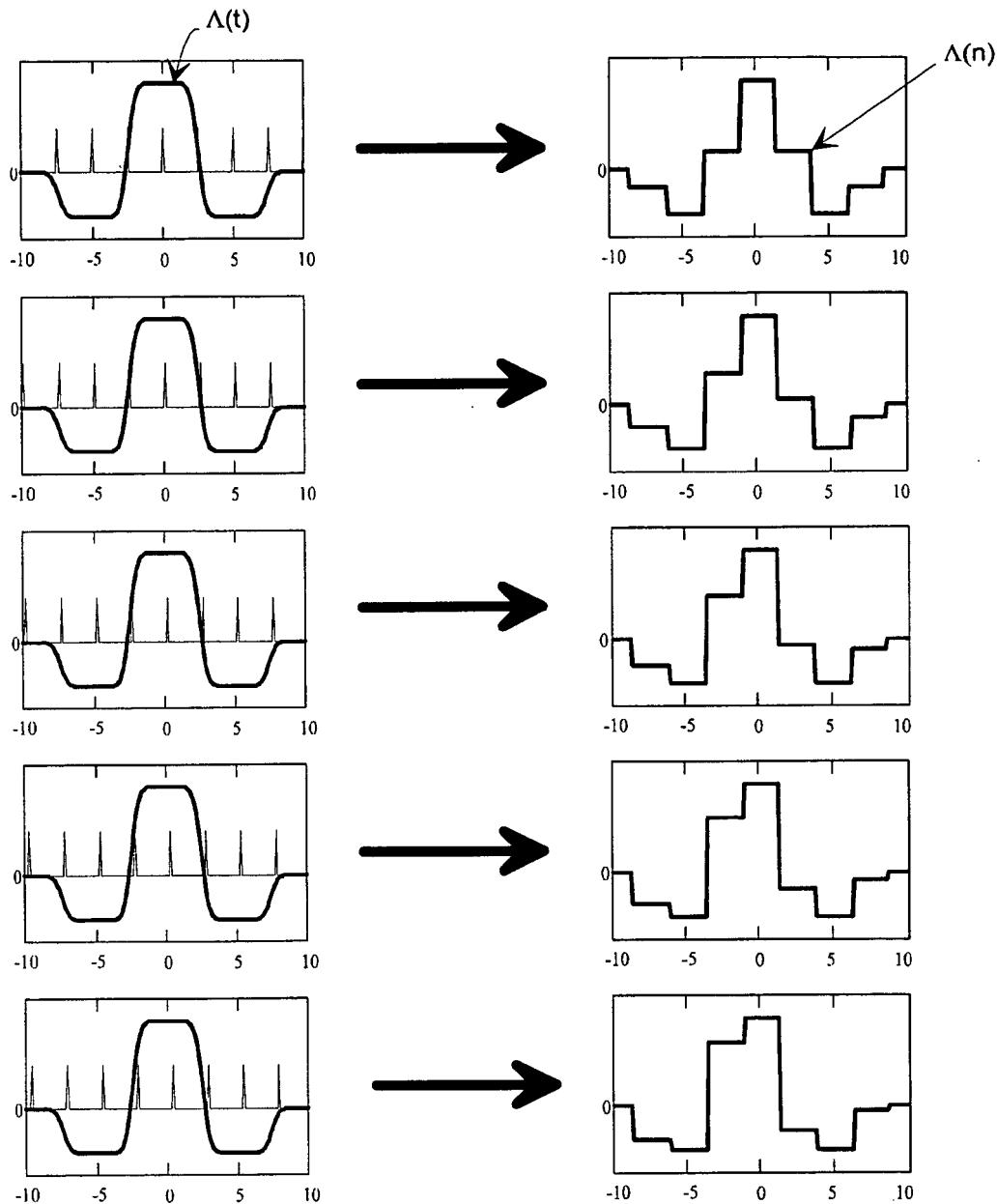
FIGURE 10. The trace labeled Λ(t) on the left-hand side is the continuous time cross-correlation function. The trace labeled Λ(n) on the right-hand side is the correlation results produced by a Time-Integrating Correlator, with "time sample" separations of 125 ps. Note how the "even" samples (e.g. the peak at 0 ns) are essentially unchanged with the relative time shift of the received signal versus the reference code, while the "odd" samples change significantly with even a 125 ps time shift.
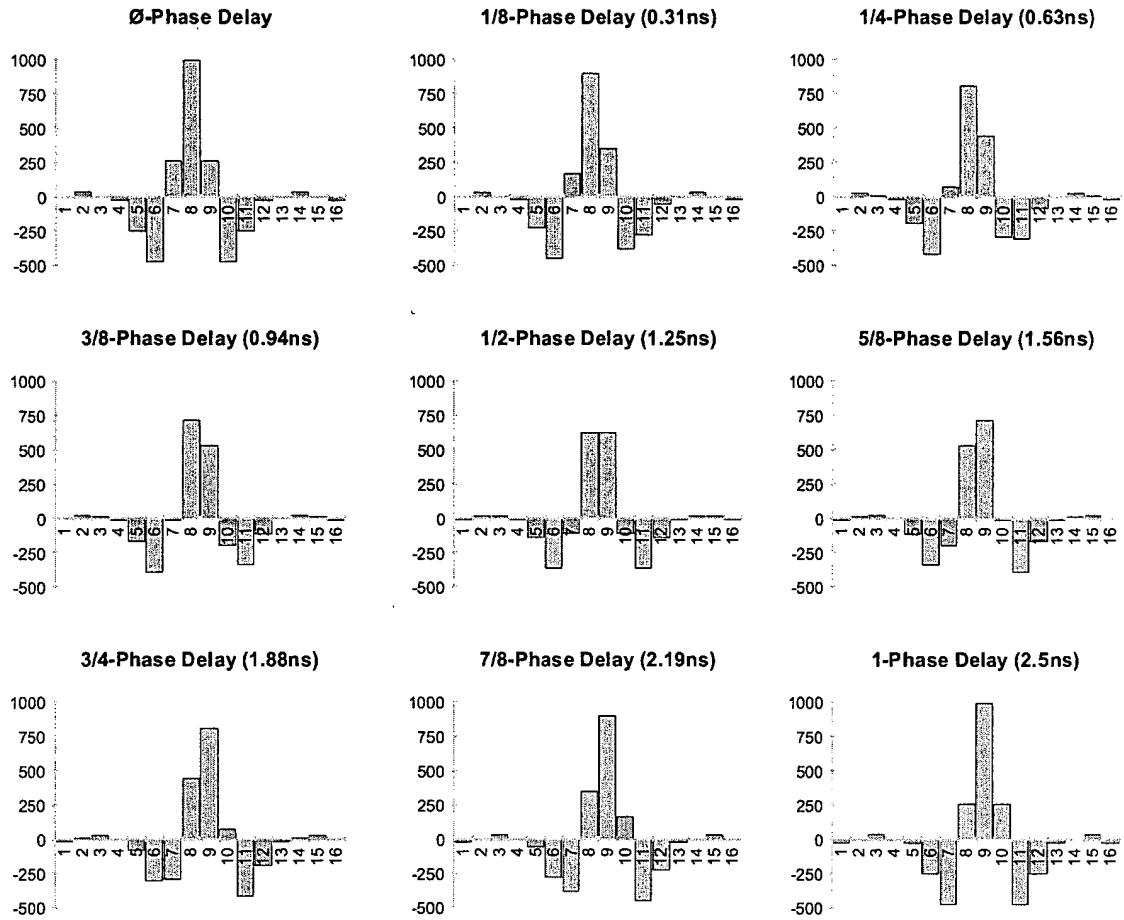
FIGURE 11. Evolution of the pattern of output phases of the Time-Integrating Correlator as the alignment of the received signal and reference code shifts in increments of 1/8-phase. These are for a 31-doublet maximal sequence with phases (*i.e.* time shift bins) separated by 2.5 ns, and each graph representing an additional 312 ps relative time shift between received signal and reference code. The pattern differences are quite easily detected using neural networks in software. Note that a time shift of 2.5 ns produces the same pattern in the lower right graph as in the upper left graph, except the pattern is centered in bin 9 instead of bin 8.

MULTIPLEXING AND ADAPTIVE POWER CONTROL

The use of correlation for reception of different code sequences allows multiple Localizers to transmit at the same time in the same area. In spread spectrum parlance this is Code Division Multiple Access (CDMA), which suffers from what is known as the *near-far problem*. Essentially, a nearby Localizer transmitting at the same time as a far away Localizer can overwhelm the distant signal. As with other spread spectrum CDMA systems, Localizers can handle this problem by adaptively reducing their transmit power when communicating with other nearby Localizers. Since Localizers communicate in bursts with low duty cycle, Time Division Multiple Access (TDMA) can also be used to deal with the near-far problem. As with Aloha Net, Localizers that find too much noise at one instant can move to another "time slot".

17

COMMUNICATION

Our cooperative ranging approach to localization requires *communication* between Localizers. To send information, code sequences have to be modulated in some fashion. The simplest technique is to use "antipodal" modulation. This means either a given code sequence or its inverse is sent to represent one bit of information. The receiver will then detect a positive or a negative correlation peak. To eliminate the ambiguity of what is a "Ø" and what is a "1", certain sequences of bits are used as a preamble, yet never appear in the message.

Other forms of modulation can also be used, such as transmitting one of several possible codes, or delaying the impulse sequence by fixed amounts in order to place the received peak in one of several possible TIC phases.

CLOCKS AND TIMING

GPS satellites carry atomic clocks that are synchronized to an absolute standard. In principle, a GPS receiver can determine the range to a satellite by comparing the absolute time broadcast by the satellite to a local absolute time standard. Fortunately, GPS receivers can use the signals from four satellites to determine the absolute time by finding a consistent solution. Still, GPS requires extremely accurate and stable master clocks because it is a broadcast beacon system.

Localizers can use *ordinary crystal oscillators* for two reasons:
- Localizers can exchange signals even if their clocks differ, as long as the differences do not accumulate over the short duration of an exchange. A good crystal oscillator is more than sufficient.
- In the process of exchanging signals, cooperating Localizers can determine the amount by which their clocks differ, and can compensate for the different clock rates. This is possible because the short term stability of a crystal oscillator can be much greater than its absolute accuracy.

For example, assume two Localizers have determined the relative clock rate of each other's clocks. Each Localizer can then measure range with a fractional error given by the fractional error in its clock during an exchange of signals. Within a network of Localizers, only one Localizer needs to have a very accurate clock for other Localizers to absolutely calibrate their clocks. Independently knowing the distance between two Localizers can also be used for calibrating the clocks of all communicating Localizers.

Even with a perfectly accurate clock, a Localizer will have unknown circuit delays that will affect the measured time-of-flight delay for a signal. As long as these delays are relatively stable, they can be measured and factored out of the range computation. The technique for doing this requires a Localizer to receive the same signal it sends. All the measured delay will then be due to circuit delays.

PROTOCOLS FOR SYNCHRONIZATION

Before a pair of Localizers can perform ranging transactions, they must become synchronized. Each Localizer must know when it can transmit a code and when it should be receiving a code from the other Localizer. In a typical synchronization protocol, one of the Localizers (B) would broadcast a code on a regular periodic basis. Another Localizer (A) would have to do an exhaustive search to receive one of these beacon codes. Because the reception window (the span of the Time-Integrating Correlator phases) is small, synchronizing can take on the

18

order of several seconds. Once a beacon is found, Localizer A can "track" the following beacons. Because the clocks on the Localizers are presumed to be slightly different, Localizer A must analyze the time difference between pairs of beacons and compare this to the expected beacon period. This allows Localizer A to calculate the ratio of its clock rate to the clock in Localizer B. This clock ratio can be used to correct for the inaccuracy of Localizer B's clock in all following transactions.

After this analysis, Localizer A has half of the information necessary for synchronization (when to listen to hear Localizer B), but Localizer B needs to find a time when it can receive from Localizer A. After Localizer A finds the beacon, it could start transmitting a return code on a periodic basis. Localizer B would have to continuously do an exhaustive search for such a return beacon, waiting for both A's search and analysis to finish, and then waiting for its own search to complete. When Localizer B completes the search, it can indicate synchronization has been achieved by modifying the next beacon, such as by antipodally modulating the code in the beacon. Alternately, Localizer B could continuously wait for a return code at a fixed phase in the beacon period, and require Localizer A to perform this second search as well as the first one.

Since the time between beacons will typically be on the order of a millisecond, the time to do the second half of the synchronization search can be greatly reduced by placing the return codes at a small known delay before or after the beacon code. For example, if Localizer A always transmits the return code immediately after receiving the beacon, then Localizer B will only have to search a small time span after the beacons to find the return codes. Assuming that Localizers have a range less than a kilometer, only the first 3 microseconds of the millisecond beacon period must be searched. This decreases the second search of the synchronization process from seconds down to tens of milliseconds. Also, knowing that Localizer A turned the beacon back immediately, Localizer B can calculate a round trip time and use that to generate an approximate distance to Localizer A. In a slightly different protocol that makes Localizer A do most of the work, Localizer B always listens for a return code immediately before each beacon, and Localizer A is required to search backwards from the beacon to find the correct time to hit Localizer B's fixed reception window. This protocol allows A to calculate the first approximate distance instead of Localizer B. As mentioned above, Localizer A will know when the search is complete when Localizer B responds by modifying the next beacon.

PROTOCOLS FOR COOPERATING LOCALIZERS TO DETERMINE RANGE

Figure 12 shows the simplest ranging transaction. Ranging requires sending a code from a first Localizer to a second one, and then getting another code back. Note that unlike RADAR, the return signal is not the echo, but a retransmission of a new code. In Figures 12, 13, and 14, a positive pulse on the timeline represents the transmission of a code, and the reception of a code is represented by a negative pulse. Localizer A transmits a code at time T1 in Figure 12, a time which is known to hit a prearranged reception window centered around time T2 on Localizer B's timeline. The shaded boxes in the figures represent calculation times, when the processors on the Localizers are analyzing the results of receptions. Localizer B must do a calculation to determine the actual arrival time of the code from Localizer A, which may not have arrived exactly in the center of the reception window. This calculated actual time is labeled as time T2a. The protocol for ranging includes a fixed time delay, called ΔTc, which is large enough to allow a Localizer to determine the actual

arrival time, plus a small safety margin. Localizer B delays this ΔTc time after the actual arrival time T2a, and sends the return code at a time called T3. Localizer A knows the value of the ΔTc delay, and has an approximate distance to B (to within ±10 meters) from previous synchronization procedures. This allows Localizer A to calculate an arrival time and schedule a reception window centered around this time, called T4. Then Localizer A must calculate the actual arrival time, T4a, during a calculation time represented by a gray box on A's timeline. The difference between time T4a and T1, minus the known ΔTc delay, is the round trip propagation time of the signal between the two Localizers. Half of this propagation time, times the speed of light, is the distance between the two Localizers. The calculations are complicated somewhat by the fact that the clock rates of the two Localizers are assumed to be different. But an approximation of the ratio of the speeds of the clocks of the two Localizers is available (from previous synchronization procedures) and this can be taken into account.
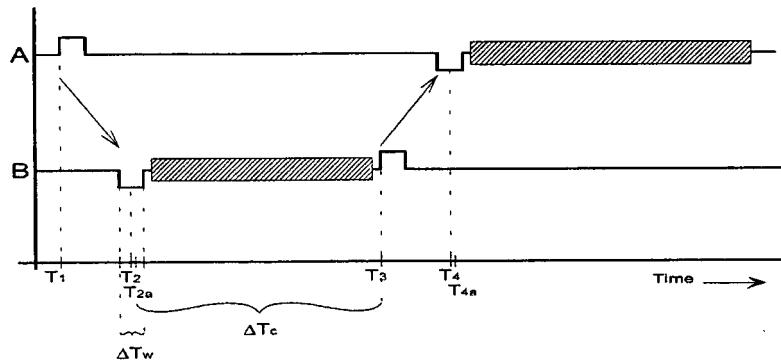

FIGURE 12. Timeline representation of simple ranging protocol. A positive pulse on the timeline represents the transmission of a code, and a negative pulse represents the reception of a code.

Figure 13 shows an improved ranging protocol. The total time between all measurements is decreased, making it less susceptible to drift in the clock rates of the two Localizers. Also, a layer of digital communication is shown, indicating how the accuracy can be increased when the Localizers share information. Localizer A sends a code at time T1 in Figure 13, which Localizer B expects to receive around time T2. A small fixed delay ΔTd after the expected reception window, but before taking the time to analyze it, Localizer B sends a return code. Localizer A schedules to receive this return code at a time T4 that is calculated from the approximate distance and the known value of ΔTd. Both Localizers then calculate the actual arrival times, T2a and T4a, of their respective received signals. If Localizer A's transmission did in fact arrive exactly at time T2, Localizer A could calculate the round trip propagation time from the difference between time T4a and time T1, minus the known ΔTd delay.

Localizer B's return transmission time was based on the expected reception time, T2, instead of the actual reception time T2a. The difference between these two times is calculated, and transmitted from Localizer B to Localizer A as a series of codes after the two ranging calculations are complete. These data codes are transmitted a fixed delay time ΔTc after the start of the return transmission, which gives Localizer B time to do all the calculation necessary. When Localizer A has received this time difference, it can be used to correct the round trip propagation time to take the actual arrival time at Localizer B into account. Note that Localizers A and B have different clocks, running at perhaps slightly different rates, and the "actual arrival time" on either Localizer's clock would be meaningless. Only the

difference between times on one Localizer's clock is ever sent to the other Localizer. Even this time difference must be corrected for known clock rate ratios in a complete algorithm.
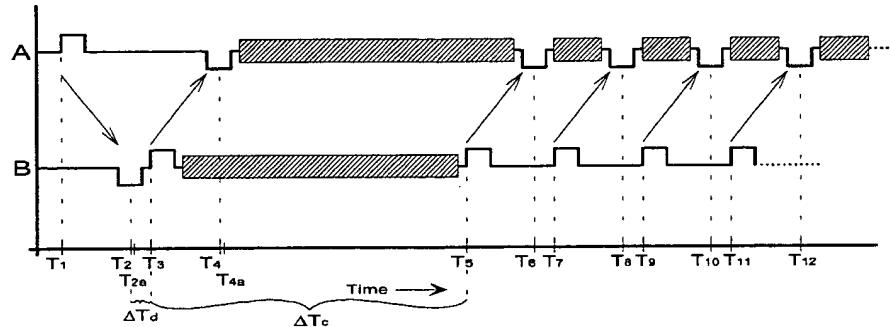

FIGURE 13. Timeline representation of an improved ranging protocol, which measures a round trip delay with quick turn-around by Localizer B and post-correction of the predicted reception time after an accurate calculation of the real reception time by Localizer B.

Figure 14 shows a more complicated, and thus more complete, ranging transaction protocol. Localizer A sends a ranging code to Localizer B, which replies immediately with a return code. Both Localizers analyze the codes to calculate the actual arrival times. Then the Localizers reverse roles, and Localizer B initiates a similar ranging exchange: Localizer B sends a ranging code, which Localizer A receives and immediately returns. Again, both Localizers analyze the codes to calculate the actual arrival times. Finally, Localizer B sends its results to Localizer A as a series of codes containing digital bits of information. The information sent would include the difference between the expected and actual arrival time of the first ranging code received at Localizer B, and the round trip time of the second complete ranging exchange. With this information, in addition to its own measured values, Localizer A has enough information to solve two equations for two unknowns, and calculate both the distance and the ratio of the clock rates between the two Localizers.
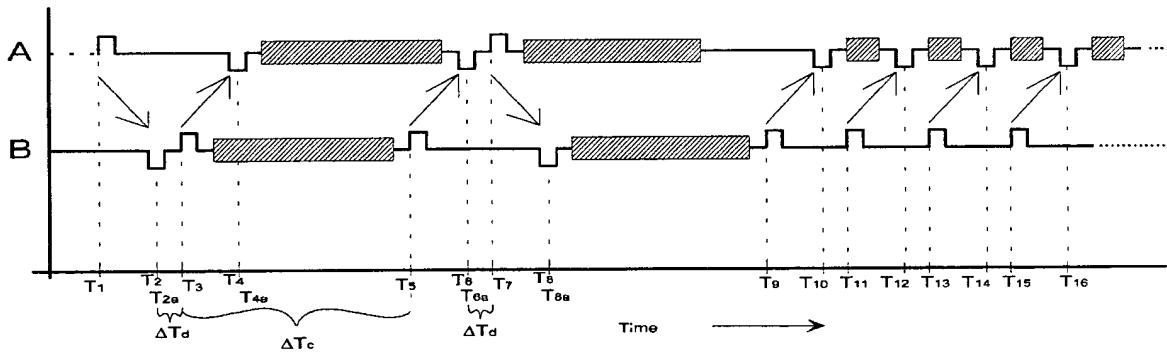

FIGURE 14. Timeline representation of a more complicated ranging protocol, again using quick turn-around and digital post-correction.

HARDWARE OVERVIEW

We have taken a *systems approach* to the design of Localizers. We perform only the most time-critical and processing-intensive functions in hardware, and bury most of the system complexity in software. As much as possible, the sequential operation of the hardware is controlled by a general purpose processor. This way, the details of operation can be worked out over time, and can be quickly changed if there are unanticipated problems. For transmission, the basic task is sending coded sequences of Gaussian impulses at precise times. For reception, the basic task is correlating the antenna signal against a known code sequence with precise alignment in time.
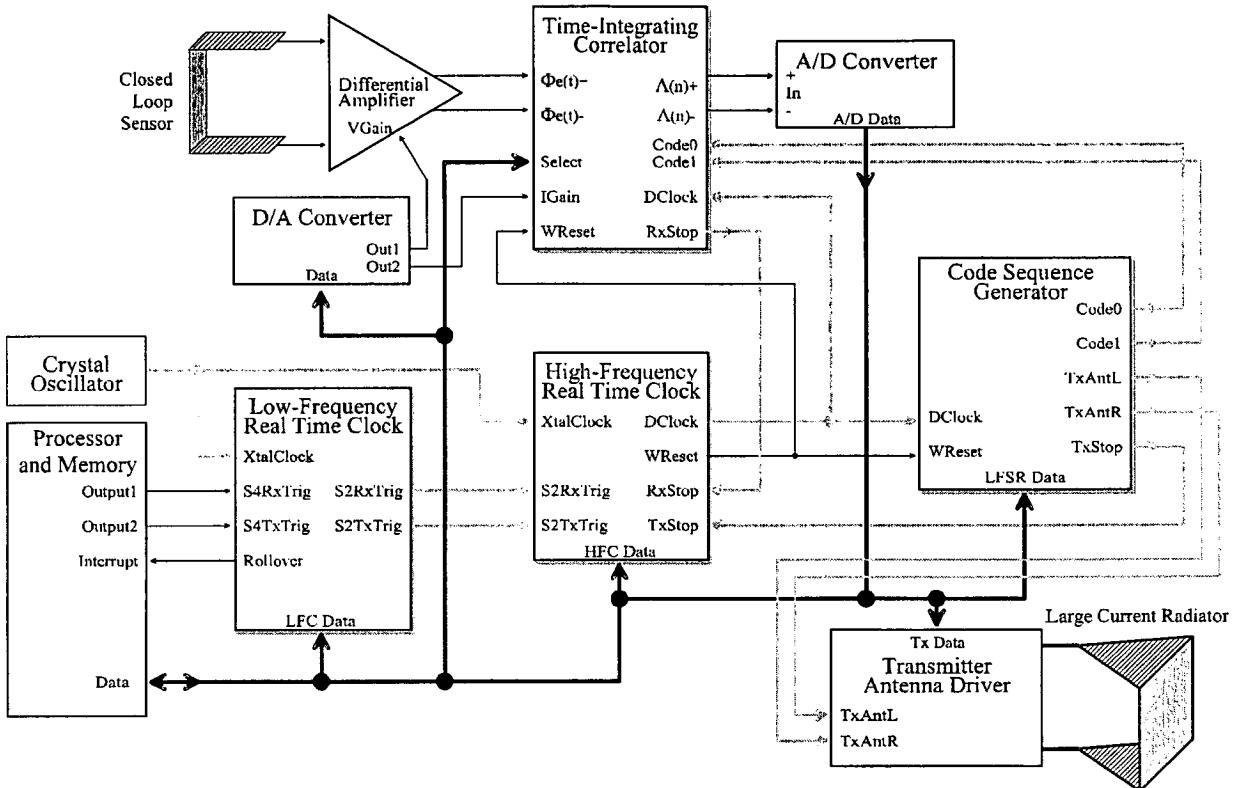
FIGURE 15. Block diagram of the Localizer hardware. Shaded lines represent low noise differential signals, and shadowed boxes represent circuits implemented with low noise logic.

REAL TIME CLOCK

To precisely trigger an event in time to 30 ps resolution, one would ordinarily need a 33 GHz clock and a counter capable of operation at this frequency. This would require Gallium Arsenide logic and be very power consumptive. Our approach obviates this problem by using the very high short term stability of crystal oscillators and the repeatability of conventional logic. The idea is to subdivide the period of a crystal oscillator into much finer graduations. In fact, this is done in several steps.

We start with a crystal oscillator running continuously at a frequency that is not too power consumptive (nominally 6.25 MHz). The signal from this oscillator is used to constantly clock the coarse resolution part of a Real Time Clock counter. A few milliseconds before a scheduled event, a voltage controlled ring oscillator (VCO) running at 200 MHz is phase-

locked to the crystal oscillator. The divide-by-N counter in the feedback path of the phase-locked loop (PLL) is used as the high resolution part of the Real Time Clock counter. The 5 ns period of the VCO is further divided by selecting a tap around the ring, and by using a programmable delay generator with 30 ps (or finer) resolution.

PHASE-LOCKED LOOP

The Phase-Locked Loop (PLL) is composed of the following components:
- Voltage Controlled Oscillator (VCO)
- Phase comparator
- Charge pump
- Loop filter
- Divide-by-N counter

Except for the divide-by-N counter, these components have their own supply pads, and they are in separate islands surrounded by guard rings which also have separate supply pads.

The VCO is an 8-stage ring oscillator implemented with the same differential current-steering buffers as are used for the other low noise circuits. The buffer delays are controlled by varying the tail currents through the differential pairs. As such, the VCO circuit includes a voltage-to-current converter that linearizes the voltage-to-frequency response of the ring oscillator.

The phase comparator is a conventional digital phase-frequency detector with Up and Down outputs for controlling a charge pump. It is implemented with single-ended current-steering NOR gates. The charge pump uses differential pairs that switch current sources between the output and an internal node[7]. The internal node is biased at the same voltage as the output to minimize signal dependent charge injection when the current sources are connected to the output. Simulations have shown that the phase error versus output charge transfer function is smooth and continuous through the origin. In other words, there is no deadband, as is usually the case with a digital phase comparator.

The loop filter is implemented as a lag-lead network on-chip using a polysilicon resistor and poly1/poly2/metal1/metal2 capacitors over N-well. The second order capacitor is 200 pf in order to close the loop with a natural frequency of 12.6 KHz using 1 uA of output current from the charge pump. Despite its size, the loop capacitor is kept on chip to prevent "wow and flutter" in the VCO frequency due to different on-chip and off-chip ground potentials.

The divide-by-N synchronous counter in the feedback path of the PLL also serves as the high frequency segment of the Real Time Clock. It can be set to divide by 16, 32, or 64, for operation of the VCO at 200 MHz with either a 12.5 MHz, 6.25 MHz, or 3.12 MHz crystal reference.

LINEAR FEEDBACK SHIFT REGISTER

The same type of pseudo-random noise (PRN) code sequence generator is used for both the transmitter and receiver sections. The design is a fully programmable 25-stage LFSR (Linear Feedback Shift Register). It is a universal code generator capable of generating Maximal Sequences (up to 2^{25}-1 bits long), Kasami Small codes (256-member families up to 65,535 bits long), Kasami Large codes (32,800-member families up to 1023 bits long), Gold and Gold-like codes (4097-member families up to 4095 bits long), and more. To use the code sequence

generator, the processor has to initialize registers with the polynomial value, the seed value, and the sequence length.

Gold codes, for instance, are often described as the XOR of two Maximal Sequences of length 2^n-1, which can be generated by an LFSR of size n. The 2^n+1 different members of a Gold code family, for a given pair of Maximal Sequences, are formed from the 2^n-1 different alignments of the Maximal Sequences, plus the two sequences themselves. It turns out that a single LFSR of size $2n$ can generate Gold codes of length 2^n-1 by using the polynomial which is the product of the polynomials for the constituent Maximal Sequences. The particular Gold code which is generated depends on the seed value used to initialize the LFSR. In other words, when the product polynomial is used, the LFSR will be stuck in one of 2^n+1 subcycles. This same scheme works for Kasami codes using the product of two or three polynomials.

For maximum performance, the 25-stage LFSR is implemented in the Galois configuration. In this arrangement, each input D_n is either output Q_{n-1}, or $Q_{n-1} \oplus Q_{25}$, as determined by the polynomial coefficient for the particular code. Other than the CMOS latch for the polynomial coefficient and the seed bit, the logic for each stage is merged into a single complex differential current-steering flip-flop.

To further minimize switching noise, the clocks to the flip-flops in the LFSR are staggered. This tends to smooth out the residual current surges drawn from the supplies. The clock signal ripples through 5 buffers, and the output of each buffer is further buffered to drive 5 stages of the LFSR. The earliest clock signal is used for the last stage of the LFSR. The output of the last stage is latched for one-half clock cycle before being fed back to earlier stages. This prevents the feedback signal from changing before the clock signal reaches the earlier stages.

TRANSMITTER SECTION

The antenna-driving output of the transmitter section looks like a standard H-bridge, as is commonly used to drive stepping motors (Fig. 16). Each bit in the code sequence determines whether the current initially flows one way or the other through the bridge. The current through the antenna is turned **on** by connecting opposite sides of the antenna to VDD and VSS. The current is turned **off** by connecting both sides of the antenna to the same supply, VDD *or* VSS, which means there is always a closed path for current to flow. In other words, we generate a step change in current through the transmit antenna by causing a step change in voltage across the antenna. Our experiments have proved that this works. In effect, the "radiation resistance" makes the transmit antenna look like a resistor (of about 2 ohms) while the voltage is switching.

To generate an impulse doublet, the two sides of the transmit antenna are first switched to different "supply" voltages and then switched to the same "resting" voltage. Ideally, the resting voltage is midway between the supply voltages, so that the average voltage of the transmit antenna does not have to change. Ordinarily, this would require a positive and negative supply with a common ground. With a single supply, the resting voltage has to be either VDD or VSS. We have incorporated in the antenna-driving circuits four options for selecting the resting voltage. Two of these options seek to balance how often the resting voltage is VDD and how often it is VSS.

For switching current through the transmit antenna, MOS transistors work better than bipolar transistors (even though their current gain is less than bipolar for a given size), because they can be turned off as rapidly as they are turned on. Since it is only the *switching edge speed* that is important, we can cascade a series of exponentially-sized CMOS inverters to drive the final stage. The actual delay through this driver chain is not important as long as it is stable and repeatable, and the final stage is driven as fast as possible.

Each of the four arms of the 'H' bridge has two programmable delay elements (one for each edge) to adjust making and breaking the switch connection in the respective output transistor. This can compensate for the different switching delays of the P-channel versus N-channel transistors, and for other circuit mismatches. Experiments have shown that there can be considerable "ringing" in the transmit antenna, which does not happen when the making and breaking of the switch connections are carefully aligned.

Each arm is also divided into multiple ($\geq$ eight) sections which can be individually enabled for power control. Multiple bond pads are allocated to the driver outputs to minimize bond wire inductance. The P-channel and N-channel output transistors are isolated from each other by guard rings and physical separation to prevent latchup. Substantial on chip capacitance is interwoven with the drivers to ameliorate the effect of the bond wire inductance for the supply connections.
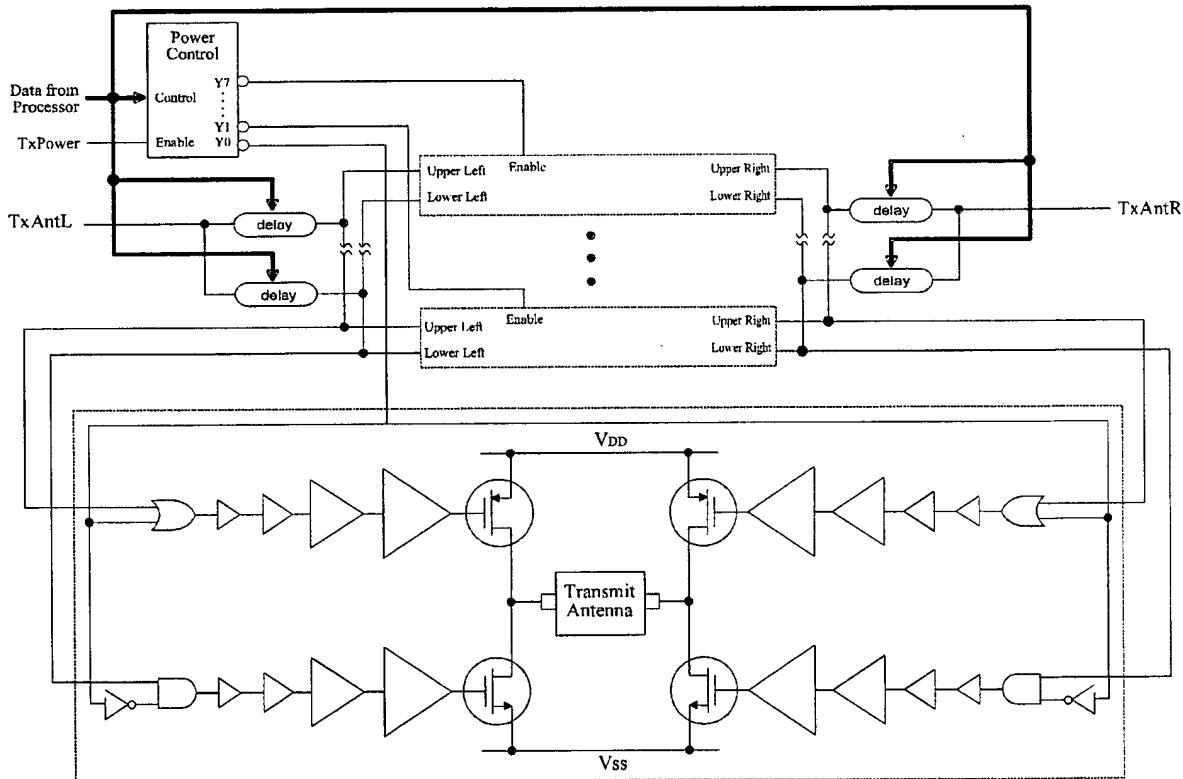

FIGURE 16. Block diagram of the transmitter antenna drivers. A typical section is shown in detail to illustrate the 'H' bridge configuration of output transistors which drive the transmit antenna. There are eight to twelve sections which can be separately enabled for adaptive power control.

RECEIVER SECTION

The receiver section includes the receiving antenna amplifier, the code sequence generator, and the Time-Integrating Correlator (TIC) with 32 integrators   The receiving antenna amplifier and the code sequence generator feed the Time-Integrating Correlator (TIC) with the amplified received signal and the reference code respectively.

A high gain amplifier with sufficient bandwidth to handle 1 nanosecond impulses is difficult to design in current CMOS technology. The maximum bandwidth is achieved by having a cascade of multiple stages with a small amount of gain and wide bandwidth per stage. The overall gain is the product of the gain of each stage. However, the overall frequency response is also the product of the frequency response for each stage. So there is an optimum gain per stage, which based on simulations is not surprisingly about 2.7 (*i.e.* e).

In 2μ CMOS we have designed and tested a DC-coupled differential gain block with variable gain up to 6 dB, flat frequency response to 300 MHz, and unity gain at 900 MHz. In 1.2μ CMOS we have designed and simulated a DC-coupled differential gain block with 10 dB gain and flat frequency response to 800 MHz. In these simulations we were able to optimize the transistor parameters so that a cascade of 6 stages had 62 dB of gain with ~1.6 dB of peaking in the frequency response. To actually achieve this performance in silicon, the frequency response of individual stages must be adaptively controlled.

The receiver section uses the same circuitry as the transmitter section for generation of the code sequence at a nominal 100 MHz chip rate (200 MHz Gaussian impulse rate). The same code sequence is fed to each integrator in the TIC, but delayed by a different amount for each "phase". A phase corresponds to an integration window 5 ns wide, but the phases are spaced 2.5 ns apart, so that their windows overlap. The integrators are implemented using a differential transconductance amplifier with a floating poly1/poly2 capacitor across the outputs. Multiplication by the code sequence is done by multiplexing the two sides of the differential received signal to the differential inputs of the integrator. To multiply by zero, the integrator inputs are muxed to the same mid-voltage. Due to the finite impedance of the current sources in the integrators, the integrator outputs have to be sampled at the end of the code sequence. The Sample-and-Hold (S/H) circuits serve this purpose, and also serve the function of buffering the integrator outputs. The outputs of the S/H circuits are muxed onto a common analog bus to be digitized by the A/D converter, which is read by the processor.

NOISE REDUCTION

Our Localizer technology was under development for three years prior to the start of sponsorship by ARPA in January of 1994. We had lab bench Localizers made with our own custom CMOS chips and other commercial chips on PC boards. An explicit goal of this project is integration of our circuits with the functions performed by the commercial chips into a single chip. Our experience with these PC board Localizers shows that a single chip Localizer is not only desirable, but necessary to reach the performance specifications we have set. These are *sub-meter* range accuracy over *100 meter* distances in real time for this phase of the project, and eventually *centimeter* range accuracy over 1 *kilometer* distances.

Currently, the primary limitation on the distance we can communicate is self-generated noise. This noise reduces the signal to noise ratio (SNR) of the received signal and affects the

26

stability of the timing circuits, *e.g.* the phase-locked loop (PLL). Significant noise sources include:

- VDD and VSS currents from the 200 MHz digital logic flowing off-chip and coupling into the receiver antenna.
- PC board logic signals that switch during the 10 μs transmission and reception window
- Digital logic in the correlator

Two conventional ways of dealing with self-generated noise are shielding and physically separating the front end of the receiver from the noisy circuits. Our experience is that shielding is particularly *ineffective* against extremely broadband noise being picked up by extremely broadband circuits. Using an antenna preamplifier that is physically separated from the noisy circuits (*i.e.* on a separate chip) defeats the goal of having a single chip transceiver.

Self-generated noise is an inherent problem for nonsinusoidal communication systems, because the correlator and logic circuits have to operate with edge speeds and clock rates that are in the same frequency range as the actual transmitted and received signals. Sinusoidal systems (such as GPS) avoid this problem by doing all their digital processing at a much lower frequency than the carrier frequency — the received signal is immediately down-converted to an intermediate frequency or baseband.

We face the problem of self-generated noise head-on in this project in several ways:

- By migrating the circuitry that has been on the PC board *on-chip*, we eliminate the high current signals and large dipole moments.
- In place of conventional (noisy) CMOS logic circuits, we use current-steering and other low noise logic for those circuits which must operate during the critical 10 μs reception window.
- We turn off (*i.e.* stop the clock) for all conventional CMOS logic (including the processor) during the critical reception window.
- For those circuits which must operate while transmitting (including the PLL), we use noise immune differential logic circuits.

The current-steering low noise logic we use is inherently power consumptive, although less so than static CMOS logic operated at 200 MHz. Static CMOS logic can use very little power but is inherently noisy. This tradeoff dictates that Localizers operate under two regimes with some overlap when transitioning from one to the other. During the reception window, the low noise logic is powered and operational, and all clocks to the static CMOS logic are stopped. All other times, the static CMOS logic is operational, and the low noise logic is powered down. Fortunately, current-steering logic can be powered down by turning off its current sources.

LOGIC CIRCUITS

We use four types of logic. In addition to conventional static CMOS logic, there are both single-ended and differential current-steering logic, and differential series pass logic. Differential current-steering logic is well suited to making flip-flops. A D-type flip-flop with Set/Reset inputs and Q/Q* outputs requires only 24 transistors. A comparable flip-flop in static CMOS typically requires 35 transistors. Also, additional logic functions can be merged into these flip-flops with only a few more transistors. For instance, a two-input multiplexer feeding the D input adds only 4 transistors.

We have discovered that conventional CMOS logic can be efficiently combined with differential current-steering logic when the conventional logic is used for effectively static

signals. For example, polynomial coefficient values are loaded by the processor into static latches that configure the LFSR. These are never changed while the LFSR is operating.

We use 5-bit DAC's to set the bias currents for the current-steering logic. This allows us to characterize the performance of the current-steering logic, to adjust the speed/power tradeoff in different sections, and to compensate for process variations.

SOFTWARE DEVELOPMENT

DEMONSTRATION SYSTEM

The original code for the "lab bench" Localizer demonstration system received both clock and synchronization signals over an fiber optic cable from a PC. After the demo code was switched over to the new development system, enhancements were made to do the synchronization in software. Much like a totally wireless Localizer will have to do, the receiver now searches for a beacon signal from the transmitter. This enhancement allowed us to gain some experience writing more complete protocols for Localizers in a real-world environment.

TOOLS

For chip design, we use several different programs to enter schematics and simulate the circuits (PSpice), lay out the circuits (ICED), and compare the IC layouts to the schematics and check for design rule errors in the layout (ICED:DRC and DRACULA). Unfortunately, there are no hard and fast standards for file formats for these programs, and they cannot quite read each other's formats. Originally, hand work with a text editor was necessary to convert schematic files from PSpice into a form that the DRACULA program could use for schematics vs. layout checks. As the chips and circuits became more complex, this hand editing task became too cumbersome, and a program to automate the task was written. (This program is called GARLIC to compound the whimsy in the names of some of the commercial programs.) At first this was a simple filter that converted name fields from one form to the other. But over the last year and a half, additional functionality was added bit by bit. This program now evaluates expressions, extracts sub-circuits from libraries, checks for several common errors, and handles multi-page schematics and off-page node references. We are now able to use more of the features of our schematic capture program, and still export them to the layout vs. schematic checking program.

Several automatic placement/routing packages were evaluated for their potential ability to expedite the cumbersome process of custom physical design, including analog cell development, (*i.e.* physical transistor placement) block placement, and routing.

Our largest effort went to evaluating the package which held the most promise, Carnegie Mellon's **Acacia**. It features transistor placement and automatic wiring for analog cells, and attempts to minimize parasitics and wiring length. Unfortunately, it lacked features we needed, (such as the ability to layout cells for wiring by abutment) and the cells it produced were more than twice the size of what we could do by hand (with significant reduction in performance).

28

RANGING SIMULATOR

To allow software to be developed in parallel with the hardware, a software simulator was written. The environment chosen for this was the Microsoft Windows™ operating system on a PC. Each simulated Localizer is a separate task under Windows, and the Windows message queuing system is used to simulate the transmission and reception of code sequences. A special task (called the Aether) acts as a clearinghouse, calculating transmission times between Localizers, and keeping track of event times. This program informs each Localizer task whether or not its reception events were successful, and when it is free to schedule another transmission or reception. Using a multitasking operating system like this allows us to spawn reasonably large numbers of copies of Localizer tasks (several dozen) and debug problems with synchronization and collisions.

To use this system, a small set of routines was written to look very much like the low level routines we will use to talk to the actual hardware. Such routines include "load the code sequence generator Linear Feedback Shift Register," "schedule a transmission event," and "sleep until triggered". Using these calls, code to implement simple protocols was written that will be re-compiled for the Motorola 683xx computers that we have chosen for the portable demo hardware.

Under this environment, the protocol layer of code for the first Localizer demo has been written. Both synchronization and simple ranging protocols have been written and debugged. The synchronization is done with a "beacon" Localizer that broadcasts a special code on a periodic basis. In "searcher" mode, Localizers hunt for a beacon, then switch to a simple ranging protocol. The current code finds distances to the nearest meter, but does not yet use any of the advanced protocols to get greater accuracy.

All these tasks (Aether, Beacon, and Searcher) display their current status in small text windows on the Windows screen. The Aether task logs Localizers being turned on and off, and displays the known simulated distance (a piece of information denied to the Localizer tasks). The Beacon tasks indicate when they get an acknowledgment of their beacon, but do not have enough information to calculate any ranges. The Searcher Localizers display their current mode, and the range estimate when available. A typical test would be to run several Beacon tasks and one Searcher. When the Searcher finds and ranges to one of the Beacons, that Beacon task is terminated. The Searcher detects loss of the ranging codes, recovers, goes back into synchronization mode, and finds the second Beacon.

The current simulation does not simulate the effect of noise. The code sequence and arrival time of signals is the only information currently communicated. Simulations of the effect of time domain noise on the correlators have been done, and this information will be used to add noise to ideal correlation values. This will be used to debug peak detection algorithms in the Localizer code.

TESTING OF TRANSMITTER CHIP (AETHER1/2)

A test chip (**aether1**) was submitted for fabrication by Orbit Semiconductor in their 1.2 µ, double-poly CMOS process in July 1994. The major components in this chip include:
- The transmitter antenna drivers and control circuits;
- A 200 MHz phase-locked loop (in low noise logic);

- A 25-stage LFSR (Linear Feedback Shift Register, for pseudo-random pattern generation) in low noise logic, with programmable seed and polynomial registers;
- A 16-bit run-length counter in low noise logic, with preload register for the LFSR;
- The two highest speed portions of the Real Time Clock (in low noise logic);
- Twelve 4-bit DAC's for biasing the low noise logic and controlling 4 edge-delays;
- Extensive built-in test circuits for externally probing and driving various subcircuits;
- Processor interface including clock-gating and synchronizing.

DEVELOPMENT SYSTEM

A system for exercising the **aether1** chip was designed and fabricated. It consists of several parts:
- A motherboard with power regulation, D/A and A/D converters, fiber optic link, and interface logic;
- A COB (chip-on-board) daughterboard for the **aether1** chip, plus regulators for voltage and current biases;
- A commercial daughterboard with a Motorola 68332 processor, RAM, and ROM.

This development system was designed to be used like the one that controls our existing lab-bench Localizers. Code is written in C, cross-compiled for the Motorola 68332 processor on a PC, and up-loaded over the fiber optic link. A "smart" terminal emulator runs under Microsoft Windows. It adds a layer of symbolic debugging on top of the ROM debugger built into the MC68332 development board. Using the Windows OS allows us to communicate with several development boards at the same time in different terminal windows, as well as running the cross-compiler in its own window.

The COB daughterboard was fabricated with polyamide so that the **aether1** chip could be epoxied in a cavity and gold wire ball-bonded directly to the board. This eliminates the lead inductance of a chip package and minimizes the wire bond inductance, especially for the transmitter driver outputs. The transmitter antenna is clamped directly to gold-plated contacts on this board.

FABRICATION PROBLEMS

On 12 September 1994, we were notified by Orbit Semiconductor that they had scrapped the run for the **aether1** chip we submitted in July due to their own processing problems. On 22 September, we were notified that they had scrapped the backup run for the **aether1** chip. Fortunately, we had submitted a slightly revised version of this chip (**aether2**) in August, but the net result was that we did not see first silicon on this design until 19 October.

For noise isolation, we designed the **aether2** chip with five sets of separate supply pins for different sections of the chip. Of the twelve chips we received from Orbit, every one had at least one of the five separate sets of supply pins shorted together. The problem was not in the design, because we were able to exercise all of the different sections on at least some of the chips. By forcing current through the shorted supply pins, we were able to identify the cause of the shorts as contacts to poly2 over poly1. These contacts are perfectly permissible by MOSIS scaleable rules as supported by Orbit, but apparently their process wasn't up to specifications, since the poly2 contacts were punching through the thin oxide separating poly1 and poly2. Even for the sections on particular chips that did not have their supplies shorted, our testing results were questionable, since many of our circuits used isolated poly1/poly2 capacitors.

We had Orbit test the 24 instances of **aether2** that were on their backup wafers. The results were the same. At least one of the five separate sets of supply pins was shorted on each of the chips. In consultation with one of their process engineers, we also learned that they were modifying our mask layouts in a manner that could lead to design rule violations. To match MOSIS scaleable rules to their design rules, they were performing bloats of certain mask layers in a naïve manner. In addition, they were doing high temperature metal processing which forced metal1 busses to "squirt" through vias to metal2, thereby causing shorts on the metal2 layer.

We would have switched to another fabricator for our chips, but Orbit Semiconductor is the only choice for multi-project wafer runs in a 1.2μ analog CMOS process with floating capacitors. Instead, we modified our design rules (and our physical layout) to compensate for their processing problems. One, we disallowed contacts to poly2 over poly1. Two, we bloated our mask layers the same as Orbit, and then checked for spacing violations using Orbit's design rules. Since this is a more conservative check, our layouts are still scaleable, and can be fabricated by other foundries that may support MOSIS scaleable rules.

ANTENNA DRIVERS

Even though we were not able to feed the antenna drivers from the code sequence generator on the **aether2** chip, we were able to separately test the antenna drivers using externally generated waveforms. The first result was that our latchup protection measures proved sufficient. From voltage and current measurements, we were able to estimate the driver sink resistance at 0.6Ω and the source resistance at 1.3Ω, which compares favorably with our simulations. Dynamic measurements into 1Ω loads showed currents up to 2 amps, rise times of 650 ps (through the P-channel drivers), and fall times of 500 ps (through the N-channel drivers).

ANALOG TESTING METHODOLOGY

As part of the design of the **aether1/2** chips, numerous BIST (built-in-self-test) circuits were incorporated for externally probing and driving various subcircuits. We found several problems with this approach. First, it was very time-consumptive to modify existing cells to include BIST, and the physical layout of the entire chip took much longer. Of more import, it was difficult to probe the low noise logic, and even more difficult to probe internal nodes without significantly reducing performance — even when not being monitored. We chose to build in sense amplifiers that could convert the low level (< 1 volt) differential low noise logic signals to standard CMOS levels. These amplifiers can be powered down for low noise operation, or powered up for probing "in circuit" signals at nearly full speed operation. This approach may be needed for volume manufacturing, but it proved inappropriate at this stage because we really needed to measure analog voltages and currents. These include bias voltages and currents and differential and common mode signal levels.

From our experience with testing the **aether2** chip, we developed a new methodology for testing our designs. The first principle is: don't try to probe the internals of a working system. Instead, make copies of the subsystems on the same chip, probe the interface signals of these subsystems; then iterate until the smallest circuits are being probed. When the chip is tested, if a large subsystem checks out, there is no need to analyze its subsystems. But if a large

subsystem does not work, then the problem can be traced to its smaller components. Other goals were:

• Allow for probing and driving analog voltages and currents.
• Allow circuits to be interconnected.
• Make all control programmable.
• Make the physical layout and wiring simple.

Using these principles, we came up with a system whereby signals could programmatically be switched using transmission gates (controlled by bits in a large serial shift register) to 16 different analog bus lines that connect to bonding pads. The analog bus lines are arrayed in two groups of 8 lines on either side of a row of circuits to be tested. The circuits are simply dropped in and wired to nearby transmission gates, which are connected to individual analog bus lines. The choice of which analog bus to use is done in the layout by dropping a via on the respective metal2 bus line. By setting the serial shift register bits under software control, we can interconnect, drive, and probe the desired circuit(s). We call this combination of analog bus lines and serial shift registers a "rail". If a circuit being tested needs a digital input signal, then a serial shift register bit can be used directly.

The compromise with this approach is that circuits cannot be tested at full speed because of the loading from the rails and the package. In cases such as testing a VCO, local buffering of the output can allow the circuit to operate while observing the output, albeit at reduced levels. In other cases, we have used the rails to setup and control a circuit, and used dedicated pads to drive and/or probe the circuit.

Using this approach, we have sped up our design and layout process significantly, and have submitted for fabrication one or more test chips per month since February 1995. This has allowed us to incrementally design, lay out, and test on a rolling basis all of the circuits that go into a complete transceiver.

DYNAMIC SHIFT REGISTER

The heart of this system for building test chips is a 12-transistor serial shift register, partially dynamic and partially static, that is a mere 44 x 20 μ in size, and needs only two control signals plus a serial in and out. The circuit shifts dynamically using a single phase **Clock** signal. There is also a **Keep** control signal. When asserted, **Keep** transfers the shift data to the register output and switches to a static hold mode. In other words, the output of the shift register does not change while shifting. This prevents conflicting settings of the transmission gates and circuit inputs controlled by the shift string. In addition, by setting **Keep** high and **Clock** low at power-on, the shift registers are reset to output low.

TESTING SYSTEM

We considered using commercial A/D and D/A boards in a PC for controlling the testing of our chips, but we found none that could be configured to programmatically probe, drive, or disconnect each chip pin. Since A/D and D/A IC's are now available with multiple channels and serial data inputs and outputs, we built our own interface board for testing chips. It connects to a PC over a parallel printer cable, and provides serial data to the conversion chips, address and control signals, and signals for loading the serial shift string inside our test chips.

In-house software had to be written to test chips on this system. To avoid errors in describing the chip to the test software, we wrote a program that reads the PSpice schematic files

32

directly, and produces a description of each test chip. With a description of a component, the technician uses a second testing program to drive the A/D and D/A converters. This test program can drive the pins with values from vector files in traditional digital testing methodologies, or run pins through analog sweeps, or combinations of the two. Interactive commands can be used to execute the component, or a long series of test values can be written to files. These output files can be pasted into mathematical analysis programs, such as Mathcad, to analyze the results, plot graphs, and print reports.

TRANSMITTER CHIP (NSTEST11)

Last year, our development plans called for expanding the **aether2** chip to include receiving circuits in addition to the transmitting circuits already on chip. We decided instead to put the transmitter antenna drivers on a separate chip (**nstest11**, Fig. 2) for several reasons:

- We wanted to electrically isolate the Phase-Locked Loop from the noise generated by switching high currents through the transmit antenna.
- Our experience with chip-on-board showed that we could reasonably wirebond between two chips.
- We wanted to preserve the aspect ratio of the transmitter drivers on the **aether2** chip which minimized the wire bond inductance driving the transmit antenna.
- The number of signals needed to go between chips is minimized if the transmitter circuitry is divided just after the transmit waveforms are generated and before they are amplified to drive the antenna.
- The antenna drivers do not need floating capacitors, so they can be fabricated using a single-poly process such as HP's 1.2μ, available through MOSIS.

We will be able to build complete transceivers using the **nstest11** chip and the receiver chip described below. To facilitate early testing, **nstest11** has additional circuitry for generating code sequences and antenna waveforms given an external clock and start signal. This circuitry can be bypassed when **nstest11** is controlled by the receiver chip. The interface with a processor or another chip is via a serial shift protocol.

EDGE DELAY

The programmable delay elements in each of the four arms of the 'H' bridge were redesigned so that both the rising and falling edges of the transmit waveform can be adjusted separately. This allows control of making and breaking the switch connection in the respective output transistor, and can compensate for the different switching delays of the P-channel versus N-channel transistors.

For the programmable delay element, we needed a gate and a buffer that would pass limited swing differential signals. The buffer would also have to transition between set voltage levels with a linear ramp dependent on programming current. This requirement that the output saturate at fixed levels ensures that the delay is not dependent on the duty cycle of the waveform passing through the buffer.

The new buffer is a complementary differential design with diode-connected transistors that clamp the output swing. This design can be described as current diverting as opposed to current steering. While switching, the set bias current charges the output nodes until they reach full output swing, at which point the clamping diodes divert the current away.

RECEIVER CHIP (AETHER3)

We have just submitted for fabrication a chip (**aether3,** Fig. 3) with the remaining subsystems for completion of transceivers that can demonstrate sub-meter range accuracy over 100 meter distances in real time. The major components in this chip include:

- A 200 MHz phase-locked loop, in low noise logic;
- All three stages of the Real Time Clock, with the high speed stages in low noise logic, including event registers and comparators for both transmit and receive;
- Two code sequence generators, one each for transmit and receive, which includes a 25-stage programmable LFSR (Linear Feedback Shift Register), a 16-bit run-length counter, and a doublet generator, all in low noise logic;
- 32 Time-Integrating Correlator phases; each phase has a reference code latch, code/signal multiplier, integrator, offset correction, sample-and-hold, and output mux;
- A broadband differential receiver antenna amplifier composed of six cascaded stages with 60 dB gain and flat frequency response to 800 MHz;
- Eighteen 5-bit DAC's for biasing the low noise logic;
- Processor interface including processor clock gating, parallel I/O decoding, and serial protocol control.

REGULATED FSCL

We reviewed several different types of low noise differential current-steering logic reported in the literature. We chose to use Folded Source-Coupled Logic[8] (FSCL) because reports and our simulations showed that it generated the lowest noise current from the supply. From our experience using FSCL in the **aether2** chip, we learned its limitations.

The input of an FSCL buffer is an N-channel differential pair. The output is a folded load consisting of a P-channel current source from VDD and a diode-connected transistor load to VSS. When fully switched, the current from VDD flows through the output load on one side, while on the other side the current is diverted through one of the differential pair transistors. The output swing is determined by the load and the tail current. Any excess current from VDD not switched through the differential pair sets the low output level.

The low-noise performance of FSCL is achieved by operating all of the family components (*i.e.* buffers, gates, flip-flops, etc.) as class-A amplifiers. In other words, all of the current available for charging the output nodes is constantly being drawn from the supply. Decreasing the bias current to limit power consumption also reduces switching speed. Having low power consumption also limits fanout and the ability to drive long wiring loads.

For a given size load, there is a tail current that achieves the desired output swing. If the load is too small, the bias generator can increase the tail current, but only up to the point where the tail current transistor falls out of saturation. Erring on the side of making the load too large slows down the switching speed by increasing the parasitic capacitance of the load transistors, and by requiring a smaller tail current to maintain the desired output swing. Thus, inaccuracies in transistor modeling and variations due to processing and temperature can cause FSCL components to fail, making the output swing too small or the switching speed too slow.

To fix the problem with load-sizing in FSCL, we developed what we call Regulated Folded Source-Coupled Logic (RFSCL). We changed the diode-connected transistor load in the output legs of FSCL to a diode-connected transistor above a transistor operated in the linear

34

region. We kept the diode-connected transistor because it adds a threshold of level shift, and makes it easier to keep the other transistor in the linear region.

The key to making RFSCL work is the bias generator which controls the load resistance and the tail current so that the low and high level outputs are regulated to fixed voltages. Because RFSCL has an extra degree of freedom in the control of the output load, it is possible to vary the bias current over a large range, effectively varying the switching speed. This allows us to adjust the speed of the RFSCL to match the needs of the circuit it is in. (In fact, we are able to use RFSCL buffers to form the ring oscillator at the core of the phase-locked loop.)

INTEGRATOR

A critical component of the Localizer system is the integrator at the heart of the Time-Integrating Correlator. It must have both sufficient bandwidth to handle 1 ns wide impulses and high enough output impedance to leak very little charge over a 10 μs period (the length of a thousand chip code sequence). It must also have a very low input offset voltage. To achieve these goals, a number of new circuit topologies were explored.

The new integrator design is based on a folded cascode differential output transconductance amplifier. P-channel transistors are used for the input differential pair so that the common mode of the input signal can be biased low. This allows N-channel switches to be used for muxing the amplified receiver signals to the integrator inputs. Regulated cascodes are used for the P-channel current sources in the output legs because they have superior output impedance and only one threshold drop. An optimal bias for the N-channel cascode transistors in the output legs is derived from a regulated cascode current mirror. The bias for the N-channel current sources in the output legs is derived from the common mode feedback circuit. This configuration for the common mode feedback adds no additional voltage drop, thereby maximizing the differential output range.

Increasing the bias current for the integrator increases its bandwidth, but also significantly decreases its output impedance. At a bias current (20 μA) where simulations show the 1 dB bandwidth to be greater than 500 MHz, the output impedance is ~41 MΩ. At this bias, the transconductance is ~47 μ mhos. This translates to an effective "impedance gain" of about 66 dB.

Because the Time-Integrating Correlator is not operated continuously, the input offset error of the integrator amplifiers can be automatically corrected before each operation of the integrators. The input offset is canceled by creating a negative feedback loop around the integrator amplifier, connecting to a much smaller differential pair wired in parallel with the amplifier inputs. Because the feedback loop includes an integrator, the input offset error is driven to zero. The differential correction signal is maintained in a track-and-hold (T/H) circuit while the integrators are operating. Several different circuits were tried for canceling the signal dependent charge injection in the T/H, but the extra active circuitry caused more leakage over time. We settled on using a simple transmission gate feeding a capacitor, with a dummy switch for charge cancellation.

SAMPLE-AND-HOLD

Each correlator phase includes a Sample-and-Hold (S/H) circuit for maintaining its integrator output while the processor does an A/D conversion on all 32 correlator phases. A compact high gain differential amplifier with differential output was designed for implementing the S/H

35

function. The amplifier is based on a complementary folded cascode topology[9]. It achieves 70 dB of DC gain in a single stage through the use of regulated cascodes in the output. The common mode feedback is efficiently merged with the feedback which closes the S/H loop[10]. The physical layout is only 72 x 42 μ.

TRANSMISSION OF IMPULSES USING THE LCR ANTENNA

We were able to observe the received signal waveform of a sample pulse train transmitted by a Large-Current Radiator (LCR) antenna over a range of **2 – 30 feet**. This was made possible by an experimental setup whereby we ensemble-averaged 1000 or more repetitions of the received pulse train. The signal-to-noise enhancement we achieve by this process is comparable to the enhancement we expect to achieve by correlation of a 1000-impulse coded sequence (*i.e.* 30 dB).

The transmitter in this experiment used an LCR with a 2" square front face, and was made with commercial bus driver chips. We expect less ringing in the antenna signal when we repeat these experiments with our own transmit antenna drivers (*i.e.* **nstest11**). The receiver consisted of a Close-Loop Sensor[11] feeding a broadband differential input amplifier. The transmitter generated a synchronization pulse for triggering the digital storage oscilloscope over a fiber optic link. Since the transmitter was also battery powered, no conduction path existed between transmitter and receiver.

In both Figures 17 and 18, the ensemble averaged waveforms are plotted together with the vertical position corresponding to the separation of the transmit and receive antennas. In Figure 17, the amplitudes are normalized, and in Figures 18, the relative amplitudes are preserved. These plots show the evolution of the received signal with both time and distance. The wavefront is progressively delayed with increasing distance, but also note the changing shape. At close range, the signal reflects the waveform of the current through the LCR, which was a square wave with some ringing. As the receiver is moved out of the near field of the transmitter, the received signal becomes proportional to the derivative of the current through the LCR, (*i.e.* the body of the square wave disappears).
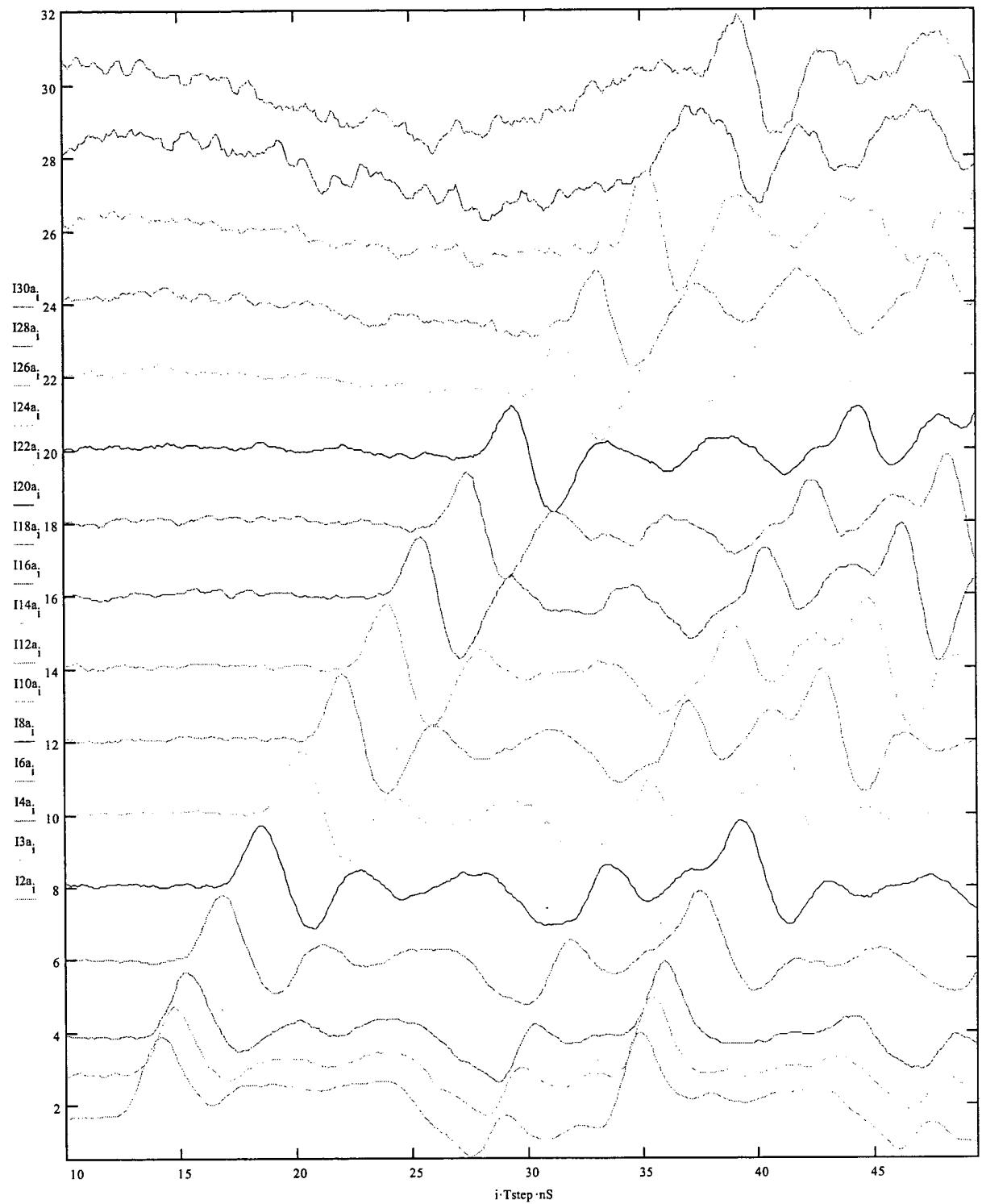
FIGURE 17. Digitized Oscilloscope traces of Impulse streams received at distances of **2 feet thru 30 feet** in 2 foot increments, with their amplitudes normalized. Their placement in the graph is proportional to the range. The waveforms have been box-averaged with a 500 ps period. The impulses were transmitted in doublet pairs (+impulse followed by a –impulse 13 ns later) with a doublet separation of 20 ns. The scope traces were synchronized via a trigger delivered by fiber optic cable, with up to 1023 traces internally averaged by the scope.
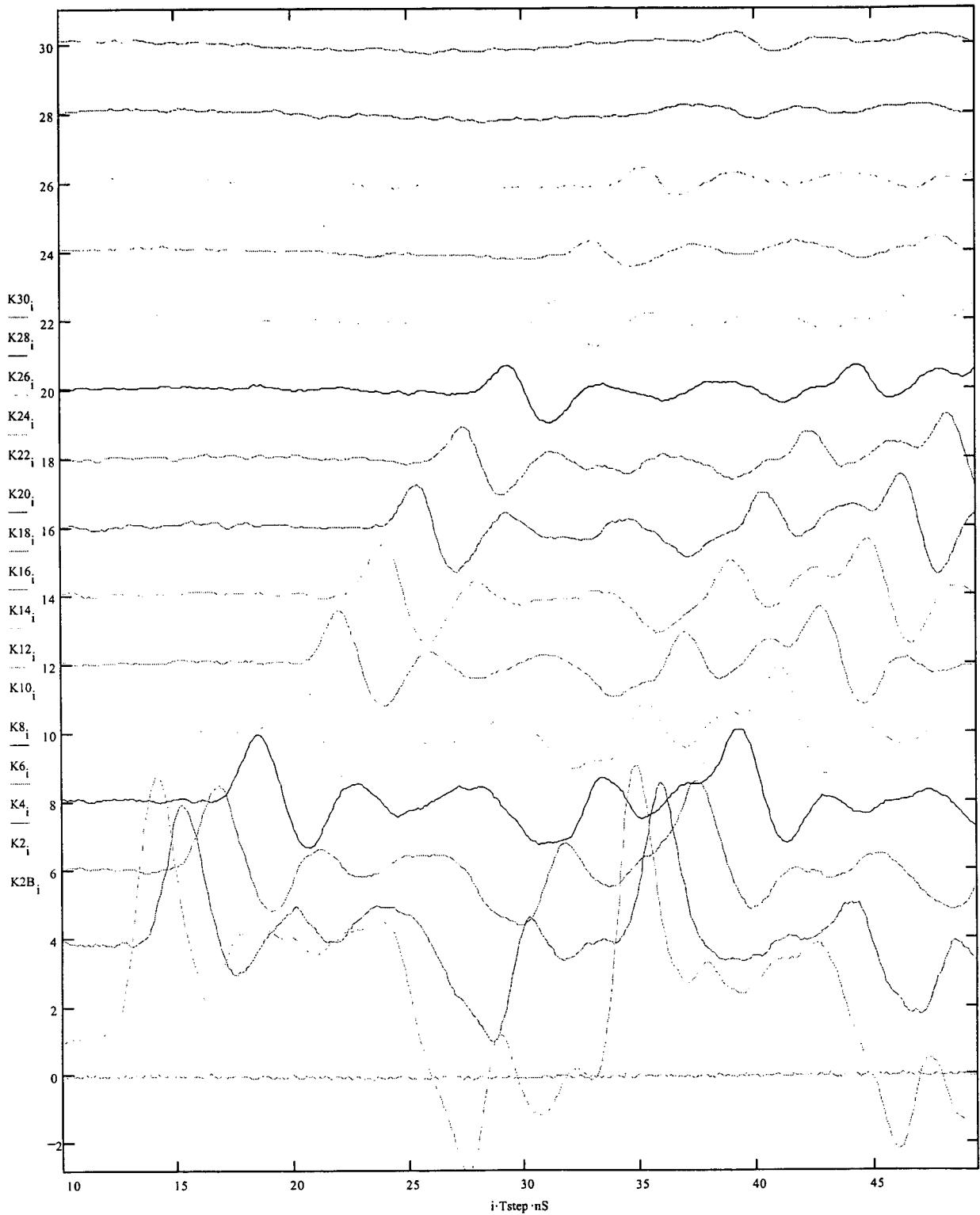
FIGURE 18. Digitized Oscilloscope traces of Impulse streams received at distances of **2 feet thru 30 feet** in 2 foot increments, with their relative amplitudes preserved to show the reduction in signal strength with range. The bottom trace is a baseline measurement with the transmitter turned off, showing the noise in the reception apparatus. Other than the baseline, the trace placement in the graph is proportional to the range. The waveform data are the same as in Figure 17.

PERSONNEL SUPPORTED

Principal Investigators:

Robert Fleming

Cherie Kushner

Researchers:

Robert Bruce

Mike Higgins

Steve Brittain

Staff:

Marge Windus (50% time)

Technical Point of Contact:

Robert Fleming
Æther Wire & Location, Inc.
5950 Lucas Valley Road
Nicasio, CA 94946
Tel: 415-662-2055
Fax: 415-662-2056
E-mail: bob@aetherwire.com

Administrative Contact:

Patrick Houghton
Æther Wire & Location, Inc.
5950 Lucas Valley Road
Nicasio, CA 94946
Tel: 408-450-2720
Fax: 415-662-2056
E-mail: patrick@aetherwire.com

REFERENCES

[1] "Defence Technology", The Economist, 10 June 1995, pp. 5-20.

[2] M. Mitchell Waldrop, "Research News: Fast Cheap, and Out of Control", *Science*, Vol. 248, No. 248, 25 May 1990, pp. 959-961.

[3] David H. Freedman, "Invasion of the Insect Robots", *Discover*, March 1991, pp. 42-50.

[4] Robert D. James and Atul S. Walimbe, *An Infrastructure Controlled Cooperative Approach to Automated Highways*, Intelligent Transportation Society of America, Fifth Annual Meeting and Exposition, Washington, D.C., March 1995.

[5] Henning F. Harmuth, *Radiation of Nonsinusoidal Electromagnetic Waves*, Academic Press, New York, 1990.

[6] B. E. Burke and D. L. Smythe, Jr., "A CCD Time-Integrating Correlator", *IEEE Journal of Solid-State Circuits*, Vol. SC-18, No. 6, December 1983, p. 736-744.

[7] K. Kurita, T. Hotta, T. Nakano, and N. Kitamura, "PLL-Based BiCMOS On-Chip Clock Generator for Very High-Speed Microprocessor", *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 4, April 1991, p. 587.

[8] Sailesh R. Maskai, Sayfe Kiaei, and David J. Allstot, "Synthesis Techniques for CMOS Folded Source-Coupled Logic Circuits", *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 8, August 1992, pp. 1157-1167.

[9] Richard E. Vallee and Ezz I. El-Masry. "A Very High-Frequency CMOS Complementary Folded Cascode Amplifier", *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 2, February 1994, pp. 130-133.

[10] Klass Bult and Govert J.G. M. Geelen, "A Fast-Settling CMOS Op Amp for SC Circuits with 90-dB DC Gain", *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 6, December 1990, p. 1382.

[11] Henning F. Harmuth, *Antennas and Waveguides for Nonsinusoidal Waves*, Academic Press, New York, 1984, pp. 121-123.